Using AI and Statistical Techniques to Correct Play-by-play Substitution Errors

by

Steven Wu

B.Sc., Carleton University, 2015

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2017

Approval

Name:	**Steven Wu**
Degree:	**Master of Science (Statistics)**
Title:	***Using AI and Statistical Techniques to Correct Play-by-play Substitution Errors***
Examining Committee:	**Chair:** Dr. Jean-François Bégin Assistant Professor

Dr. Tim Swartz
Senior Supervisor
Professor

Dr. Dave Campbell
Supervisor
Associate Professor

Dr. Oliver Schulte
External Examiner
Professor
School of Computing Science

Date Defended: 26 May 2017

Abstract

Play-by-play is an important data source for basketball analysis, particularly for leagues that cannot afford the infrastructure for collecting video tracking data; it enables advanced metrics like adjusted plus-minus and lineup analysis like With Or Without You (WOWY). However, this analysis is not possible unless all substitutions are recorded and are correct. In this paper we use six seasons of play-by-play from the Canadian university league to derive a framework for automated cleaning of play-by-play that is littered with substitution logging errors. These errors include missing substitutions, unequal number of players subbing in and out, substitution patterns of a player not alternating between in/out, and more. We define features to build a prediction model for identifying correct/incorrect recorded substitutions and outline a simple heuristic for player activity to use for inferring the players who were not accounted for in the substitutions. We define two performance measures for objectively quantifying the effectiveness of this framework. The play-by-play which results from the algorithm opens up a set of statistics that were not obtainable for the Canadian university league which improves their analytics capabilities; coaches can improve strategy leading to a more competitive product, and media can introduce modern statistics in their coverage to increase engagement from fans.

Keywords: Classification; Artificial intelligence; Play-by-play; Basketball

Acknowledgements

Thank you to my senior supervisor Dr. Tim Swartz for the opportunities, support, and patience throughout my time at Simon Fraser. Your research presentation at my first conference attended, CUMC 2012, about optimal time to pull a goalie is what sparked my interest in sports analytics, and this spark has been the catalyst for everything good that has happened in my academic and professional life.

Thank you to Dr. Luke Bornn for the opportunities to work with cutting edge data and methodologies. Being able to participate in your class, lab, and reading groups has pushed my abilities as a researcher. Thank you to Dr. Dave Campbell for always grounding our statistical computing class to real life applications, and for being an advocate for all the important things outside of the traditional classroom that will help us succeed after graduation. Thank you to Dr. Oliver Schulte and Dr. Jean-François Bégin for taking the time to be part of my committee. Thank you to Dr. Steve Thompson and Dr. Boxin Tang for being friendly and thoughtful in your lectures. Thank you to Charlene Bradbury, Kelly Jay and Sadika Jungic for being amazing support for our department.

Thank you to the friends I've made out here - you are all a great bunch. Special mentions go out to Khalif Halani, Matthew van Bommel, Trevor Thomson and Jennifer Parkhouse, for the great friendships fostered over the many friendly-but-overcompetitive boardgames nights. Thank you to my friends back home for making my return trips home feel special.

Thank you to my sister who has always been a role model and mentor in my life. Your IQ, wit, and accomplishments have always given me something to be proud of and strive toward.

Thank you to my mom and dad who supported me since day one. You trusted me enough to make my own life decisions and have been there for absolutely anything I needed every step of the way. You don't ask for anything and you give everything, which is a truly remarkable quality and it inspires me more than you know.

Thank you to Anja Radakovic, my #1 everything, for taking a leap of faith to move across the country with me. I couldn't have asked for a better partner by my side to adventure with these past two years. My thanks for you could easily fill up a whole page. I can't wait for the next chapter in our life together.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivating Problem

The sophistication of analytics in the basketball community is at an all-time high due to the availability of spatio-temporal data in the National Basketball Association (NBA) that is driving innovation. The MIT Sloan Sports Analytics Conference has annually showcased innovation and research in sports analytics since 2006 (MITSSAC, 2017). In recent years, a significant proportion of the cutting edge research presented is enabled by camera software that allows for tracking individual players and the ball. However, the cost of the camera software required to record this data is too high for virtually any basketball league other than the NBA and National Collegiate Athletic Association (NCAA); the last reported figure for provider Stats LLC was $100,000 per year for each NBA team (Witz, 2014). For those such leagues with budget constraints, the most common and acquirable granular data is play-by-play, which is manually recorded by scorekeepers. This method is inexpensive, as the only costs are for human labour and (optionally) software that assists the recording. For the uninitiated, play-by-play is an event log that contains details on the sequence of discrete events (e.g. steals, turnovers, shot attempts) that occur in a game of sport. Refer to Figure 3.1 for an example of basketball play-by-play.

It was not too long ago when basketball's cutting edge analytics were derived from play-by-play data; the grand prize winner for the best research paper at MIT SSAC 2010 used ridge regression on data extracted from play-by-play to arrive at an adjusted plus-minus metric for each player in the NBA that accounts for teammate and opponent strengths (Sill, 2010). Smaller leagues that do not have the budget for tracking software, but which produce play-by-play, should be able to enjoy the same level of analytical discourse that the NBA has benefitted from. Access to these analytics improve coaching strategy, roster management, and fan engagement. However, the problem of smaller budgets is a compounding one that affects the quality of the play-by-play itself: the less money that the league has to spend on the data collection, the less reliable the data is with respect to the true events that occurred in the game.

An example of this described problem of play-by-play with inconsistent quality is the data from the U Sports (formerly known as the Canadian Interuniversity Sport, or CIS) league, which will be the league analyzed throughout the rest of this project. Specifically, the relatively low budget of the U Sports league affects the following factors that have a resulting negative influence on the data's resulting quality:

- software to assist in annotation: a keyboard software that maps shortcuts for event annotations presents the opportunity for mistakes made by mistyping keys
- wages: scorekeepers are not paid like industry data entry professionals, and are usually students whose primary incentives and motivations to continue this work are their interest in the sport
- training: there is no standardized training across the keepers for each school, leading to high variability in consistency and reliability of the keepers across schools

The boxscore is an end-of-game summary of the major statistics for each player. The aggregation of statistics in a boxscore across the whole game removes key contextual information such as the score margin and players on the court. Teams in the U Sports league have interest in accessing analytics and metrics that are more advanced than ones obtained from the boxscore. One such team is Carleton University's men's basketball team, winners in 13 of the past 15 years in the U Sports league (Sports, 2017), who were interested in the impact of specific units of players deployed. Methods that can answer such a question, such as With Or Without You (WOWY) or adjusted plus-minus, are dependent upon knowing which ten players are on the court at all times during the course of the game. It was found to be impossible to get sensible results using these proven methodologies because it relied on the play-by-play's substitution logs being accurate. For this type of advanced downstream analysis, an automated solution that can "clean" the play-by-play's substitutions to guarantee five players on each side of the court at all times is necessary - one that can suggest the five most likely players on the court is ideal.

1.2 Outline

In this project, we present a novel implementation of an artificial intelligence agent which uses the contextual data surrounding the substitutions to reliably infer who is actually on the court, guaranteeing five players on the court for each team at all times. Because the goal state of our AI agent is unknown (without watching film of games to verify the correctness of the recorded play-by-play substitutions) we define two performance measures that quantify the success of our agent. Using over 6000 games from the U Sports league, we discuss the results of our framework for automated play-by-play cleaning.

The project is organized as follows. In Chapter 2 we describe the U Sports play-by-play data format, errors, and our own terminology for solving the problem. In Chapter 3 we describe agent based systems from the field of artificial intelligence and classification models from the field of statistics, and outline our algorithm for joining the two in creating our automated solution for cleaning the play-by-play substitution errors. Chapter 4 describes the method and justification for how we objectively measure the performance of our automated solution. Finally, a discussion of the results, implementation limitations, and further work is given in Chapter 5.

Chapter 2

Data

U Sports, which has 40+ participating universities across Canada for both the men's and women's basketball leagues, has published their game data online every year since the 2009-2010 season. The data collected for this paper is six seasons of the publicly available play-by-play and boxscore data. For every game, one group of workers are responsible for recording the play-by-play and another group of workers are responsible for recording the boxscore tallies. The fact that a separate party of people record minutes totals for the boxscore will be important later in Section 4.1 for validating our algorithm.

The play-by-play has four columns: timestamp, away team events, score, and home team events. Rows in the play-by-play detail the events that happened throughout the game, the time at which the event occurred, and the score as a result of the event (if it changes). Each row has an associated index that we will be referring to: the first row of the play-by-play event log is index 0, the next row is index 1, and so on. The eight types of potential recorded events are:

- goes to the bench
- enters the game
- foul
- turnover
- steal
- block
- defensive or offensive rebound
- made or missed shot or free throw

The logging of substitutions in the play-by-play data is where we see the poorest quality. A variety of substitutions problems that occur repeatedly are quantifiable from the play-by-play without needing to cross reference with game film. We enumerate these error types below and measure their frequency in Table 2.1:

Error Type 1: some games in each season have no substitutions recorded at all

Error Type 2: recording an unequal number of players entering the game versus going to the bench (see Figure 2.1 for an example)

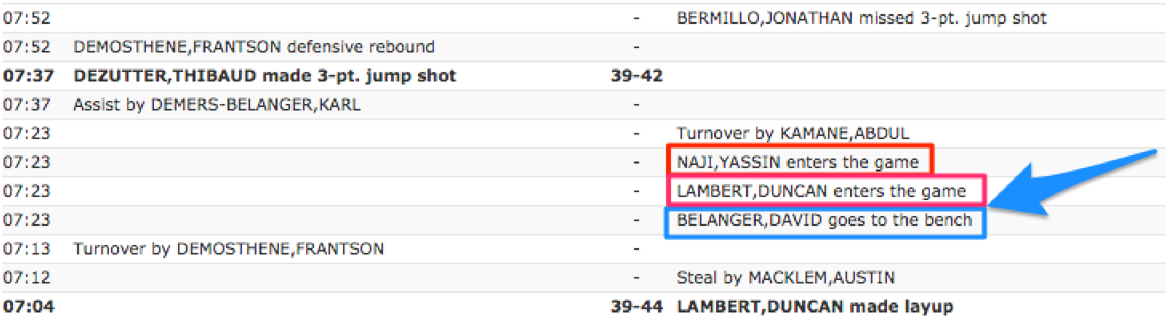
Error Type 3: player's substitution patterns not alternating between entering the game versus going to the bench (i.e. a player is marked as going to the bench, then the next substitution involving the player has them going to the bench again)

Error Type 4: recording that a substitution occurred but missing the name of the player (e.g. " enters the game") (see Figure 2.2 for an example)

Some other errors which occur can only be confirmed by cross referencing with game film; thus, these error types are not measured in Table 2.1. We list some of the errors we have observed in the play-by-play:

- recording the wrong player name in the substitution event, leading to inconsistency between the recorded substitution and the events recorded before/after (e.g. "DOE,JOHN goes to the bench" followed by "DOE,JOHN made layup")
- substitutions that happened but were not recorded (most commonly, the players who substitute between the end of one quarter and the start of the next quarter)

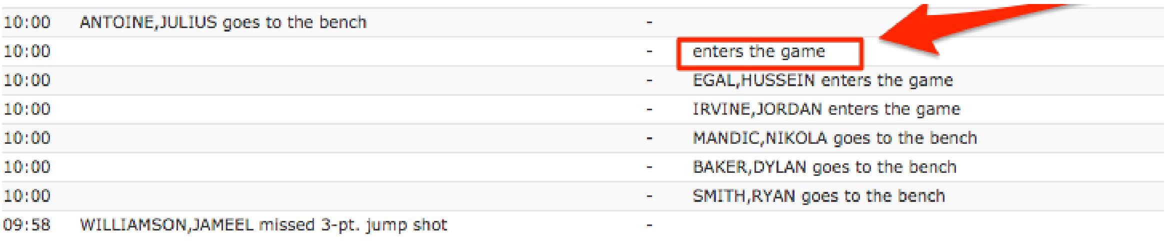
Figure 2.1: An example of Error Type 2, uneven substitutions. 2016/01/22 Laval vs. Bishop's



07:52		-	BERMILLO,JONATHAN missed 3-pt. jump shot
07:52	DEMOSTHENE,FRANTSON defensive rebound	-	
07:37	**DEZUTTER,THIBAUD made 3-pt. jump shot**	**39-42**	
07:37	Assist by DEMERS-BELANGER,KARL	-	
07:23		-	Turnover by KAMANE,ABDUL
07:23		-	NAJI,YASSIN enters the game
07:23		-	LAMBERT,DUNCAN enters the game
07:23		-	BELANGER,DAVID goes to the bench
07:13	Turnover by DEMOSTHENE,FRANTSON	-	
07:12		-	Steal by MACKLEM,AUSTIN
07:04		**39-44**	**LAMBERT,DUNCAN made layup**

To aid our discussion, we define terms that will repeatedly come up in the next sections:

Figure 2.2: An example of Error Type 4, substitutions with no names. 2014/11/21 St. FX vs. UNB



10:00	ANTOINE,JULIUS goes to the bench	-	
10:00		-	enters the game
10:00		-	EGAL,HUSSEIN enters the game
10:00		-	IRVINE,JORDAN enters the game
10:00		-	MANDIC,NIKOLA goes to the bench
10:00		-	BAKER,DYLAN goes to the bench
10:00		-	SMITH,RYAN goes to the bench
09:58	WILLIAMSON,JAMEEL missed 3-pt. jump shot	-	

- stoppage: a point in the game where play is paused, the clock is stopped, and substitutions are allowed
- substoppage: the play-by-play row index corresponding to recorded substitutions and beginnings of every period
- active play: a recorded play performed by a player that is not a substitution or technical foul

Table 2.1: The frequency of errors we can objectively identify without cross-referencing the data with video. Error Types 2, 3, and 4 are per game averages for the season.

Season	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**Avg**
# Games	798	828	829	883	911	895	905	864.14
# Error Type 1	112	48	43	23	1	1	1	32.71
Avg # Substoppages	39.71	43.75	42.45	43.68	44.45	44.64	44.30	43.35
Avg # Subs	115.3	126.2	122.9	127.4	129.3	129.5	130.1	126.0
# Error Type 2	2.87	2.75	2.70	2.66	2.69	3.09	2.30	2.72
# Error Type 3	55.46	60.88	59.12	61.00	61.29	62.12	62.35	60.41
# Error Type 4	0.000	0.002	0.008	0.003	0.015	0.102	0.107	0.035

From Table 2.1, we observe that the number of games without substitutions (Error type 1) halves in 2010, and halves again in 2012, before the problem becomes almost non-existent from the 2013 season. An additional obstacle to recover the missing substitutions for games with this error type is to also determine the time points in the game where the substitutions actually occurred. From a practical standpoint, handling games that exhibit this error type is not as important as being able to handle the other types, due to its dramatic drop off in rate of occurrence. More discussion on handling this error type is given in Section 5.2.1.

There is also a jump in both the number of substitutions and the number of substoppages when comparing 2009 to the rest of the seasons. This is most likely due to unfamiliarity and inexperience with the scorekeeping tools for recording substitutions in the inaugural year. Error types 2 and 3 are relatively consistent from year to year. The average frequency of error type 4 increases dramatically in 2014 and 2015 due to outliers [1].

[1] In the 2014 season the men's Cape Breton versus UNB game on 2015/01/23 had 85 instances, and in the 2015 season the men's and women's Thompson Rivers versus Mount Royal game on 2015/11/14 had 45 and 39 instances respectively.

Chapter 3

Automatically Cleaning Play-by-play Substitutions

3.1 Artificial Intelligence Techniques

3.1.1 Introduction to AI

In artificial intelligence (AI), the fundamental problem is to describe and build intelligent agents, which are entities with sensors to perceive its environment and actuators to perform actions on its environment (Russell and Norvig, 2009). Russell and Norvig (2009) argue that the study of AI is about rational agent design, where rational refers to the agent acting in such a way to achieve a goal based on its encoded beliefs. Numerous publications in academic and industrial literature can be found on the topic of agents; Rudowsky (2004) provides a comprehensive overview and touches on the distinction between the agent framework and other frameworks such as object-oriented programming and expert systems. Practical examples of intelligent agents in the real world include Non Playable Characters (NPCs) in video games and autonomous vacuum cleaners; both are able to perceive their environment and act dynamically within it to perform actions upon it.

The computer representation, or model, of the environment at a given point in time is called the state. A utility-based agent uses a utility function (or heuristic) to map a given state to a utility value with respect to its objective. This allows it to behave in a goal-directed manner by moving from state to state, where it chooses the next state by selecting the option with the best utility. The goal state of an artificial intelligence agent is either known (and the search is directed toward finding it) or the goal state is unknown (and the search is an exploration to try and find it, or the best possible solution).

3.1.2 Application to Play-by-play Corrections

This AI framework for thinking about problem solving is a useful one for our application. We would like to apply the tools and vocabulary of this field for the automation aspect of the task. Below we describe how we model the agent's environment and how it decides to move from state to state.

The set of periods for each game are the first, second, third, and fourth quarters, plus any overtime periods. We can break up the problem of cleaning a game into sub-problems of cleaning the substitutions of each individual period. Furthermore, for each period, we can partition the play-by-play by its substoppages. Formally, denote the whole game's play-by-play as $\mathbf{P} = \cup_{j=1}^{J} P^j$, where J is the number of periods and P^j is period j's play-by-play which contains rows that are indexed by order. For each period j, there is a set of substoppages $\mathbf{SS}^j = \{ss_1, \ldots, ss_{d_j}\}$. Recalling the definition of a substoppage in Chapter 2, every period has a substoppage at the beginning (row index 0) as substitutions are allowed at the beginning of a period before play begins; thus it is always the case that $ss_1 = 0$. Since substoppages refer to the row indices where substitutions occur, we have a mapping for period j, $\mathbf{S}^j$, that maps substoppages to the substitutions observed at that substoppage.

The initial state is the unmodified play-by-play with all of its recorded substitutions (except for ones that exhibit error type 4; those are removed as they do not provide useful information). The set of current players in the game is denoted $\omega = \emptyset$. The actions the agent can perform are removing a recorded substitution or imputing a substitution that it believes should have been recorded, at each substoppage. Each action the agent performs changes the state, as a row in the play-by-play is either removed or imputed. For each period j, the agent iterates through each substoppage $ss_k \in \mathbf{SS}^j$. At each substoppage ss_k, for each substitution $s_i \in \mathbf{S}^j[ss_k]$, we assign a confidence score based on features extracted from the contextual evidence and discard substitutions which are classified as incorrectly recorded. If there are any substitutions classified as correctly recorded, it updates ω by removing the players from ω that are recorded as going to the bench and inserting the players into ω that are recorded as entering the game. The agent uses $P^j[ss_k : end]$ (not necessarily all of it; details are in Section 3.3) to assign an activity score to every player and infer who the five most likely players are on the court for each side, where the $P[q : r]$ notation means the play-by-play from row index q until row index r. Once all of the periods and substoppages are iterated through exactly once, the agent is finished its task.

Our goal state is the play-by-play which has all of the correct substitutions recorded that reflect the substitutions in the game that actually occurred. In our case, this goal state is unknown - the only way to know it for a given game is to watch the video film, which is not a feasible task to do for every game given how many there are in a season. In our case, we want a solution which is the best approximation to the truth. How we assess our solution is discussed in detail in Chapter 4.

From surveying the data and its common errors, recordings of substitution plays are much less reliable than the active plays. Intuitively, it is easier as a scorekeeper to assign the correct player to a single action involving the ball during a live play than to correctly account for up to ten players substituting for each team. Given this, we can use the active plays as contextual data at each substoppage in the game to infer what player substitutions were most likely to have actually occurred. Specifically, for each substoppage ss_k, we have two problems that we need solutions for:

1. remove the recorded substitutions that show enough evidence of being incorrect (detailed in Section 3.2)
2. given the remaining substitutions, infer and impute the substitutions that should have been recorded, ensuring five unique players are on the court for each team (detailed in Section 3.3)

See Figure 3.4 at the end of the chapter for a visualization of the agent's workflow.

3.2 Binary Classification of Recorded Substitutions

An important task for our agent is to accurately classify whether a recorded substitution is correct or not, using the context of the play-by-play surrounding it. There are many choices of classifiers available for predicting the binary label: for example, logistic regression, linear discriminant analysis, K-nearest neighbours, decision trees, random forests, support vector machines, and neural networks (James et al., 2013).

The original motivation for cleaning these substitutions is to have the most accurate representation of which players are on the floor over the course of the game, so that advanced metrics that use this context can be as accurate as possible. A natural question from coaching staff is: how close is the resulting mutated game to the truth? For this reason, it is desirable to choose supervised learning methods that can improve with examples of play-by-play that have had the substitutions manually verified by cross referencing with video film.

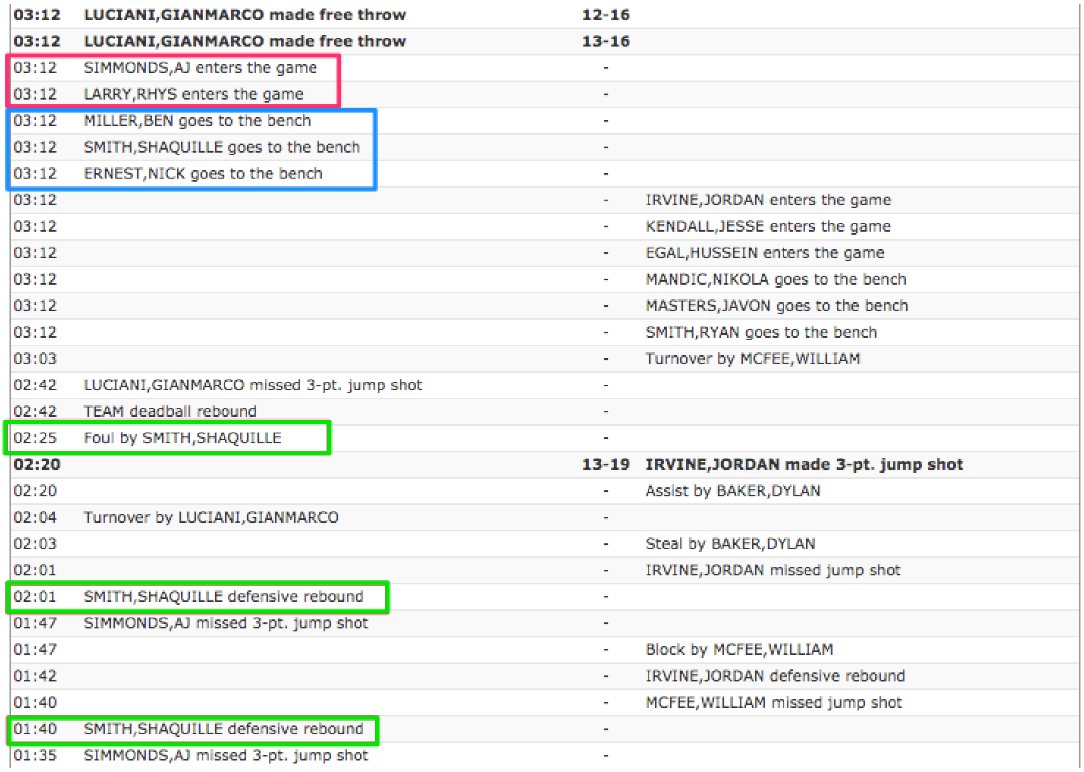
To train our classifier, we obtained five video replays of full games featuring distinct teams from the 2015-2016 men's season, which had commentators and a running score count on the video feed. For each game, we recorded which players were actually on the court for each row of the play-by-play. Knowing the true five players on the court for each side, we were able to deduce which substitutions were recorded correctly or incorrectly. Features that we believed to be predictive in whether a substitution is recorded correctly or not were collected for each substitution from the annotated games, and are detailed in subsection 3.2.1.

A recorded substitution can be one of two types: entering the game versus going to the bench. The resulting dataset has 312 "enters the game" substitutions and 311 "goes to the bench" substitutions. In terms of the balance of the two classes in our dataset, 76.1% of the "enters the game" substitutions and 76.8% of the "goes to the bench" substitutions were correctly recorded.

3.2.1 Extracted Features

For each substoppage ss_k, for each substitution $s_i \in \mathbf{S}[ss_k]$, we would like to extract features that are predictive in determining whether the substitution s_i is correctly recorded (and should remain in the play-by-play) or if it is incorrectly recorded (and should be removed in the play-by-play). For a concrete example, let us examine Figure 3.1.

Figure 3.1: An example of 11 total substitutions within a U Sports play-by-play log. 2014/11/07 UNB vs. Acadia



03:12	**LUCIANI,GIANMARCO made free throw**	**12-16**	
03:12	**LUCIANI,GIANMARCO made free throw**	**13-16**	
03:12	SIMMONDS,AJ enters the game	-	
03:12	LARRY,RHYS enters the game	-	
03:12	MILLER,BEN goes to the bench	-	
03:12	SMITH,SHAQUILLE goes to the bench	-	
03:12	ERNEST,NICK goes to the bench	-	
03:12		-	IRVINE,JORDAN enters the game
03:12		-	KENDALL,JESSE enters the game
03:12		-	EGAL,HUSSEIN enters the game
03:12		-	MANDIC,NIKOLA goes to the bench
03:12		-	MASTERS,JAVON goes to the bench
03:12		-	SMITH,RYAN goes to the bench
03:03		-	Turnover by MCFEE,WILLIAM
02:42	LUCIANI,GIANMARCO missed 3-pt. jump shot	-	
02:42	TEAM deadball rebound	-	
02:25	Foul by SMITH,SHAQUILLE	-	
02:20		**13-19**	**IRVINE,JORDAN made 3-pt. jump shot**
02:20		-	Assist by BAKER,DYLAN
02:04	Turnover by LUCIANI,GIANMARCO	-	
02:03		-	Steal by BAKER,DYLAN
02:01		-	IRVINE,JORDAN missed jump shot
02:01	SMITH,SHAQUILLE defensive rebound	-	
01:47	SIMMONDS,AJ missed 3-pt. jump shot	-	
01:47		-	Block by MCFEE,WILLIAM
01:42		-	IRVINE,JORDAN defensive rebound
01:40		-	MCFEE,WILLIAM missed jump shot
01:40	SMITH,SHAQUILLE defensive rebound	-	
01:35	SIMMONDS,AJ missed 3-pt. jump shot	-	

Focusing our attention on the Away team (second column), "enters the game" substitutions are boxed in red, "goes to the bench" substitutions are boxed in blue, and useful evidence of what substitution is incorrect is boxed in green. There are 11 substitutions; for each of them, we would like to automatically classify whether it is worth keeping, or if we should discard it and consider replacing it with what we believe to be a correct substitution. We will examine one substitution in particular, "SMITH,SHAQUILLE goes to the bench" for the Away team. A couple indicators obtained from the context surrounding the substitution indicate that it is very likely incorrectly recorded; namely, that the Away team substitutions are unbalanced (two players entering the game versus three players going to the bench) and that SMITH,SHAQUILLE is recorded to be making plays that indicate he is still in the game (fouling, rebounding).

Building on this example, we detail the complete set of features extracted for our classification model in Table 3.1. This set of predictors was obtained using domain expertise gathered from conversing with coaches, play-by-play scorekeepers, and from our own knowledge of the data.

3.2.1.1 Active Plays Features

Two features from Table 3.1 that merit more discussion are $X_{7,i}$ and $X_{8,i}$.

Intuitively, if a “enters the game” substitution was recorded correctly, we do not want to see any active plays made by the player before the substitution; the player is coming from the bench, and therefore could not have made any active plays before this point. After the substitution, active plays made by the player would be expected.

Vice versa for a “goes to the bench” substitution being recorded correctly, we do not want to see any active plays made by the player after the substitution; the player is exiting the game, and therefore would not be able to make any active plays after this point. Before the substitution, active plays made by the player would be expected.

One option to capture this notion of a player’s activity is to simply count the number of active plays seen before and after the substitution. However, substoppages sometimes soon follow the one before it, such as when a player who subbed in has quickly fouled out of the game. With that said, the context of seeing one or two active plays drastically changes depending on whether there was a lot of uninterrupted play or a little. One active play out of five active plays until the next substoppage could be a good indicator that a player is on the court, while one active play out of 60 until the next substoppage has a higher chance of being a mistake recording. A more sophisticated feature could incorporate the total number of active plays seen before or after the substitution. For simplicity and interpretability, we count the number of active plays.

3.2.2 Classification Model

We use a logistic regression model for our classifier, for the additivity and interpretability of the coefficients, as well as the probabilistic framework (which allows us to adjust the classification thresholds if necessary).

Logistic regression assumes that the observations in the dataset are independent. It is possible that the recorded substitutions to be classified do have some dependence; for example, substitutions within the same substoppage are more correlated than substitutions in another substoppage, or even another game. We have added features that capture the contextual dependency structure and believe that it is reasonable to assume that each observation in our dataset is independent.

Table 3.1: Description of extracted features at substoppage ss_k for every recorded substitution s_i; the player substituted in s_i is denoted as p_i

Notation	**Name**	**Comments**
Y_i	Correct/incorrect recorded substitution	Response variable corresponding to whether substitution s_i is correct or incorrect.
$X_{1,i}$	Difference of # in vs. # out for p_i's team	Integer from 0 to 5. Correct substitutions should have an equal number of players entering and leaving the game. The difference taken is in absolute value.
$X_{2,i}$	Total number of substitutions for p_i's team	Integer from 0 to 10.
$X_{3,i}$	Is beginning of period	Whether the substitution recorded occurred at the beginning of a period
$X_{4,i}$	Previous substitution is opposite	Whether p_i's previous substitution type is opposite to the substitution type of s_i
$X_{5,i}$	Next substitution is opposite	Whether p_i's next substitution type is opposite to the substitution type of s_i
$X_{6,i}$	Player appears more than once in substoppage	p_i can only enter or leave the game. Records of p_i doing both at the same substoppage are incorrect.
$X_{7,i}$	Plays before count	The count of player activity in the play-by-play before the substoppage (i.e.: $P^j[ss_{k-1} : s_k]$).
$X_{8,i}$	Plays after count	The count of player activity in the play-by-play after the substoppage (i.e.: $P^j[ss_k : ss_{k+1}]$)

The coefficients may vary drastically depending on the type of substitution. Because of the reasoning outlined above in 3.2.1.1, we train a separate model for each substitution type, using the same predictors.

For substitution type $t \in \{\text{enters the game, goes to the bench}\}$, the model for classifying the correctness of substitution s_i is:

$$P_t(Y_i = 1) = \sigma(\beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 X_{3,i} + \beta_4 X_{4,i} + \beta_5 X_{5,i} + \beta_6 X_{6,i} + \beta_7 X_{7,i} + \beta_8 X_{8,i})$$

where

$$\sigma(X) = \frac{exp(X)}{1 + exp(X)}.$$

If $P_t(Y_i = 1) > 0.5$ then s_i is classified as correct. The additivity of the terms is desirable for this classification problem. From the perspective of a human who would be tasked to classify these substitutions with the logic detailed in subsection 3.2.1, we are interested in applying a probability or likelihood that the substitution is indeed correctly recorded. By examining data that we extract from the context surrounding the substitution, we are either more confident or less confident in our prediction; each feature we look for in determining this prediction adds or subtracts to this probability.

3.2.3 Model Results

Regularization is a technique used in model fitting that shrinks coefficient estimates toward zero to prevent overfitting. Tikhonov regularization, also known as ridge regression or L2 regularization, is achieved by adding a component

$$\lambda \sum_{j=1}^{p} \beta_j^2$$

to the model's loss function to be minimized, where λ is a tuning parameter. This component, often called a shrinkage penalty (James et al., 2013), is small when the coefficients are close to zero. The choice of λ can be found by varying the parameter and observing how some criterion (such as classification accuracy) changes as the parameter varies. For our implementation, $\lambda = 1$.

Although we have a reasonable dataset size of over 300 samples for each substitution type, the sample of five games from which the substitutions were obtained is small and we would like to avoid overfitting to trends observed in this limited set. Training the logistic regression model without regularization led to coefficients that were too large; the most extreme example is a value of $\beta_{6,i} = -20.014$ in the "goes to the bench" model. The possibility exists where one of numerous substitutions containing the same player name in the same substoppage is recorded correctly (in other words, a substitution where $X_{6,i} = 1$ and $Y_i = 1$), however, the unregularized model would almost never be able to correctly classify such a substitution due to the extremely large negative coefficient of $\beta_{6,i}$. The coefficient in the regularized model, $\beta_{6,i} = -1.691$, makes it easier for the correct classification decision to be made (provided that there is evidence from the other features that suggest the substitution is a correct recording).

Cross validation (CV) is a commonly used technique to avoid selecting models that overfit the data they were trained on. In k-fold CV, the dataset is split into k smaller sets (also called folds) of approximately equal size. For k iterations, the model is trained on a new set of training data consisting of $k - 1$ of the smaller sets, and it is validated on the held out set. In practice, k is typically 5 or 10 (James et al., 2013). A larger k results in higher variance in your measured error and longer computational time, for the benefit of a less biased estimate of your true expected error. The benefit to this method is that for each iteration, the model is being validated on a set of data that it has not been trained on and the resulting error is a better estimate of how the model will perform in general on new unseen data. The 10 CV score in the tables are the average classification accuracies across the $k = 10$ iterations. The classification score for "goes to the bench" substitutions of 90.8% is higher than the score for "enters the game" substitutions of 84.5%, suggesting that the former substitution type is easier to classify than the latter.

Table 3.2 and Table 3.3 detail the coefficients and 10 CV classification score for our "goes to the bench" and "enters the game" substitutions respectively.

Below we interpret the coefficients for each feature for both the "goes to the bench" and "enters the game" classification models.

- $X_{1,i}$'s coefficient in both models is -0.919 and -0.718. If p_i's team's substitutions contains an unbalanced number of players entering and leaving, then intuitively the probability for any of the substitutions being correctly recorded should be lower.

- $X_{2,i}$'s coefficient in both models is -0.337 and -0.217. The larger the number of total substitutions that a scorekeeper has to keep track of, the more opportunity there is for mistakes. A maximum of 20 substitutions and a minimum of 2 substitutions can occur at a substoppage for both teams; 20 substitutions is harder to keep track of in the same fixed time allowed for keeping track of 2 substitutions.

- $X_{3,i}$'s coefficient in both models is approximately -1.8. It is not consistent practice among scorekeepers to record the substitutions occurring between periods; when they are recorded, they are either wrong or redundant (e.g. recording a player entering who was already last recorded as entering). One reason for why these substitutions are consistently incorrect is because of the time lag between the end of one quarter and the beginning of the next; when the next quarter starts, it is hard to remember who was last on the court, which leads to errors in recording the player substitutions.

- We would expect the coefficients for $X_{4,i}, X_{5,i}$ to be positive, as the most recent previous/next substitution of the opposite type is an indicator that the current one is correct; however, we see a small negative coefficient for $X_{5,i}$ in both models. This is likely due to a combination of the fact that all recorded substitutions are relatively unreliable (as noted before, a 25% rate of incorrectness was observed in our labelled data) and that the small sample size (relative to the number of games played per season) led to a dataset that was unrepresentative of the effect that we would likely see with a larger sample size.

- $X_{6,i}$'s coefficient in both models is -1.691 and -0.872. As expected, if a substoppage contains p_i as both entering and leaving the game, then the probability of substitutions involving p_i being correct should be much lower.

- For "goes to the bench" substitutions, as expected for $X_{7,i}$ and $X_{8,i}$ we see an increase in the likelihood of correctness if there are active plays made by p_i before the substitution (as it is evidence of the player being on the court prior) and a decrease if there are active plays made after the substitution (as it is evidence that the player remained on the court). The magnitude of the effect is approximately equal for both. For "enters the game" substitutions, we observe that the largest effect (amongst both models) is the negative coefficient for $X_{7,i}$ which aligns with our intuition: if we see evidence of the player on the court before the substitution, he is almost definitely not entering the game. The effect for $X_{8,i}$ is positive but very small, which aligns with our intuition that while observing active plays should be an indicator that the player has entered the game, it is common for a player entering the game to not have many actions recorded; it is in fact the least important variable in this classification model.

Table 3.2: Coefficients and 10 CV classification score for “goes to the bench” substitutions

Variable	**Value**
β_0	2.160
β_1	-0.919
β_2	-0.337
β_3	-1.881
β_4	1.295
β_5	-0.058
β_6	-1.691
β_7	0.796
β_8	-0.782
10 CV score	90.8%

Table 3.3: Coefficients and 10 CV classification score for “enters the game” substitutions

Variable	**Value**
β_0	2.690
β_1	-0.718
β_2	-0.217
β_3	-1.821
β_4	0.929
β_5	-0.303
β_6	-0.872
β_7	-2.430
β_8	0.028
10 CV score	84.5%

Figure 3.2 and Figure 3.3 show the distribution of $P_t(Y_i = 1)$, for each substitution type t, for the games in the 2009-2015 seasons in which we needed to predict the correctness of recorded substitutions. A large proportion of the density is at $P_t(Y_i = 1) \geq 80\%$. This means that the majority of the classifications made are with reasonably high confidence. Proportionally, we see extremely low density in the range $0.4 \leq P_t(Y_i = 1) \leq 60\%$ where the classification decision is hard, which is ideal.

Figure 3.2: The empirical distribution of $P_t(Y_i = 1)$ for $t =$ “enters the game”

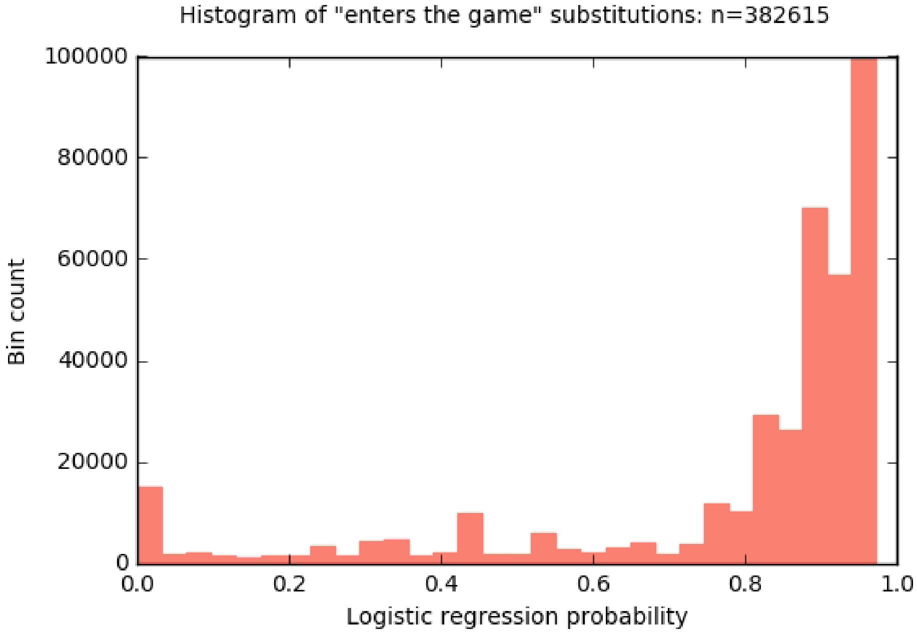
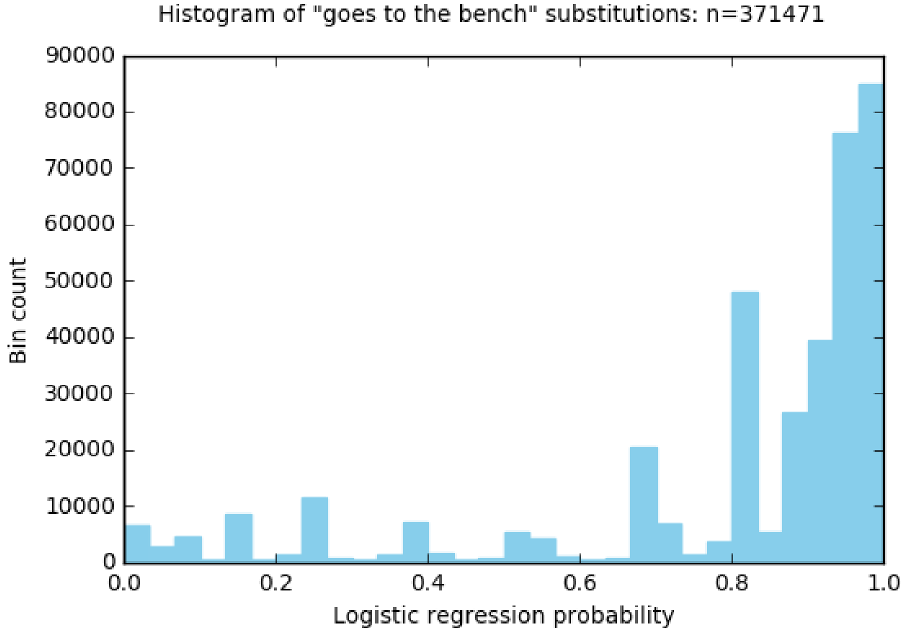


Figure 3.3: The empirical distribution of $P_t(Y_i = 1)$ for $t =$ “goes to the bench’.



3.3 Inferring Missing Substitutions

Active plays are good indicators that a player is on the court, but a lack of active plays does not necessarily mean the player is not on the court. Several reasons can explain a lack of active plays by a player who is truly on the court. As a period progresses over time, there is less opportunity for a player to make a play. There is large variance in player skill; low skilled players will not make as many plays as the best players in the league, and some players make their contributions through ways that are not tallied (e.g. setting screens, playing on-ball defense, making the right pass). As well, any player regardless of skill may play long stretches of game time without a logged event. Thus, the binary classification step detailed in Section 3.2 is important to gain information on such situations where the evidence does not make obvious who is on the court.

After classifying which substitutions are incorrect, we discard them. Applying the remaining substitutions on ω, we are left with either less than, exactly, or more than five players for each team in ω. If we have exactly five players for each team in ω the agent skips this step as there are no missing substitutions to infer.

We are interested in knowing which players are the most likely to be on the court after ss_k, for the cases where we do not have five players accounted for on the court for either team. The only information our agent can use to decide who is most likely to be on the court is the frequency of the active plays seen by each player. The agent needs a heuristic that it can use as a proxy for the degree of activity of a player. Equation 3.1 shows an activity heuristic (AH) for player p_i at a given substoppage ss_k that we use for our implementation.

$$AH_{p_i} = \sum_{r=ss_k}^{ss_k^*} \mathbb{1}_{r,j,p_i} + \alpha \sum_{r=ss_k^*}^{ss_k^{**}} \mathbb{1}_{r,j,p_i} \tag{3.1}$$

where

- $\mathbb{1}_{r,j,p_i} = \begin{cases} 1 & \text{if row } r \text{ in } P^j \text{ contains player } p_i \text{ making an active play} \\ 0 & \text{else} \end{cases}$
- $ss_{k^*} = \begin{cases} end & \text{if } ss_k \text{ is the last substoppage} \\ ss_{k+1} & \text{else} \end{cases}$
- $ss_{k^{**}}$ is the substoppage after ss_{k^*} such that the number of players $\notin \omega$ with a non-zero AH gives us enough players for the team to add to ω. Similar to ss_{k^*}, it is the *end* index if ss_k is the last substoppage.
- $0 < \alpha < 1$ is a coefficient that lowers the weight of the evidence seen in $P^j[ss_{k^*} : ss_{k^{**}}]$

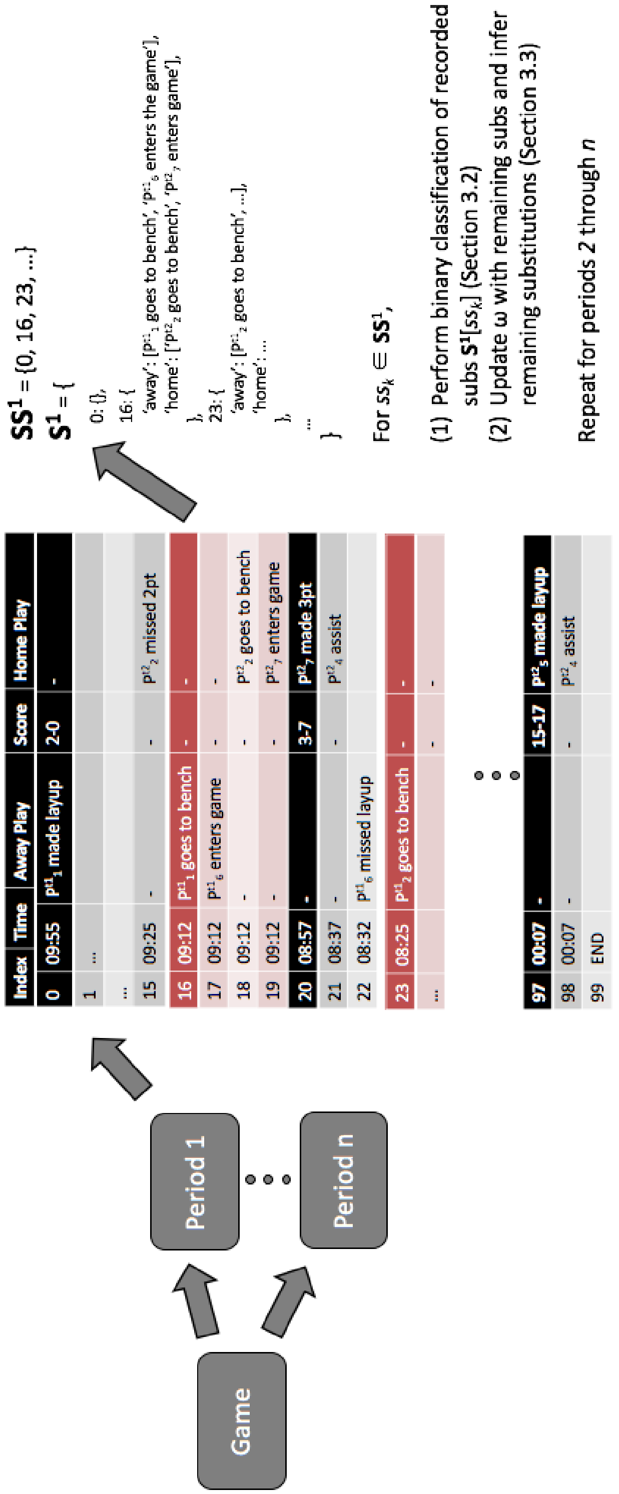
On occasion, when a team's players in ω is less than five and ss_k is too close to ss_{k^*}, $P^j[ss_k : ss_{k^*}]$ might not have enough rows to find players $\notin \omega$ with a non-zero AH. The agent will need to look ahead past ss_{k^*} to find players who are likely to be on the court that can be added to ω. The evidence from $P^j[ss_{k^*} : ss_{k^{**}}]$ is down-weighted by α to reflect the uncertainty in the substitutions that occur in the substoppages after ss_k [1].

[1] In our implementation, α is chosen to be $\frac{1}{3}$. Changing this value did not affect the results significantly, as the main purpose of it is so that evidence seen in $P^j[ss_k : ss_{k^*}]$ has a higher weight than evidence seen in $P^j[ss_{k^*} : ss_{k^{**}}]$.

If we have less than five players for a team, we add to ω in order of highest activity from the set of players $\notin \omega$ until there is exactly five. If we have more than five players for a team, we remove from ω in order of lowest activity until there is exactly five. The missing "enters the game" and "goes to the bench" substitutions naturally follow from the set difference in ω at ss_{k-1} and ω after we have performed this step.

Note: since deflections and out-of-bounds events are not recorded, not all stoppages can be determined from the play-by-play. Thus, we are restricted to inferring substitutions only at the substoppages in the game's play-by-play.

Figure 3.4: Flowchart depicting visual illustration of environment and agent's actions for a typical 1^{st} quarter where the scorekeeper does not record the substitutions at the beginning. Since there are no substitutions to classify at substoppage 0 in this case, the agent would skip that step and infer who the 10 players on the court are.



Index	Time	Away Play	Score	Home Play
0	09:55	p^{t1}_1 made layup	2-0	-
1	...			
...				
15	09:25	-	-	p^{t2}_2 missed 2pt
16	09:12	p^{t1}_1 goes to bench	-	-
17	09:12	p^{t1}_6 enters game	-	-
18	09:12	-	-	p^{t2}_2 goes to bench
19	09:12	-	-	p^{t2}_7 enters game
20	08:57	-	3-7	p^{t2}_7 made 3pt
21	08:37	-	-	p^{t2}_4 assist
22	08:32	p^{t1}_6 missed layup		
23	08:25	p^{t1}_2 goes to bench	-	-
...			-	-
97	00:07	-	15-17	p^{t2}_5 made layup
98	00:07	-	-	p^{t2}_4 assist
99	END			

$\mathbf{SS^1} = \{0, 16, 23, \ldots\}$

$\mathbf{S^1} = \{$

```
0: {},
16: {
'away': [p^{t1}_1 goes to bench', 'p^{t1}_6 enters the game'],
'home': ['p^{t2}_2 goes to bench', 'p^{t2}_7 enters game'],
},
23: {
'away': [p^{t1}_2 goes to bench', ...],
'home': ...
},
...
}
```

For $ss_k \in \mathbf{SS^1}$,

(1) Perform binary classification of recorded subs $\mathbf{S^1}[ss_k]$ (Section 3.2)

(2) Update ω with remaining subs and infer remaining substitutions (Section 3.3)

Repeat for periods 2 through *n*

Chapter 4

Results

Agent systems are evaluated on performance measures, defined as objective criteria for success of an agent's behaviour (Russell and Norvig, 2009). We define two performance measures that can objectively quantify the result of our agent cleaning the play-by-play.

4.1 Minutes Criterion

In the U Sports league, a separate party from the ones responsible for recording the play-by-play is responsible for compiling the boxscore statistics. It is important to emphasize the fact that the boxscore tallies do not result from the play-by-play itself. Since it is an account of the game from another objective party, we can use the minutes tallied in the boxscore as a ground truth to compare the minutes that are tallied from our cleaned boxscore. Taking the absolute difference from the minutes obtained from our cleaned game and from the boxscore can give us a reasonable performance measure for our algorithm's correctness, thus we take the average absolute minutes difference for each player who played in the game. The minutes criterion (MC) calculated for the k^{th} game is given by:

$$MC_k = \frac{1}{N}\sum_{i=1}^{N} |p_i^g - p_i^b| \tag{4.1}$$

where

- N is the number of players that appear in the game k's play-by-play for both teams
- p_i^g is player i's minutes tally obtained from the cleaned game
- p_i^b is player i's minutes tally obtained from the boxscore

4.2 Unknown Players Criterion

Though less frequently occurring, there is the possibility of an incorrect player name recorded for an active play. Particularly, this occurs when a lot of active plays are in quick succession and/or when inexperienced scorekeepers lose track of play.

In situations where we update ω after a substoppage ss_k and encounter a player p_i making an active play in $P^j[ss_k : ss_{k+1}]$ such that $p_i \notin \omega$, there is a contradiction that must be resolved; our model has p_i on the bench but we have come upon an event that suggests p_i is on the court. From the perspective of the agent, the true player who performed the action (instead of p_i, who it believes is on the bench) is unknown. For that reason, we call the sum of these contradictions the unknown players criterion (UPC). The UPC calculated for the k^{th} game is given by:

$$UPC_k = \sum_{i=0}^{n_k} \mathbb{1}_i \tag{4.2}$$

where

- n_k is the number of rows in the game k's play-by-play
- $\mathbb{1}_i = \begin{cases} 1 & \text{if row } i \text{ has an active play but } p_i \notin \omega \\ 0 & \text{else} \end{cases}$

For a concrete example of how this may occur, imagine an agent solution that is extremely irrational and decides for the play-by-play in Figure 3.1 that the "SMITH,SHAQUILLE goes to the bench" substitution should remain and removes another substitution. Having decided that SMITH,SHAQUILLE $\notin \omega$, the plays highlighted in green cannot have been made by him. These three instances would have the count for UPC_k for game k increase by 3.

4.3 Discussion of Results

To keep Table 4.1 compact, let

$$NAP_k = \text{number of active plays for game } k$$

$$NP_k = \text{number of players for game } k$$

From looking at Table 4.1's average number of unknown player instances and active plays per game, we see that we are unable to attribute a player that we estimate to be on the floor to an active play less than 1% of the time. This is encouraging, as detailed at the end of Section 4.2, an irrational agent would have a higher rate.

Table 4.1: Performance measures per game averages for every season for men's and women's games.

Season	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**Average**
Avg UPC_k	3.02	2.91	3.15	3.16	3.28	3.00	3.00	3.07
Avg NAP_k	376.16	380.43	377.24	374.42	375.34	373.84	377.12	376.37
Avg MC_k	3.34	3.24	3.27	3.25	3.35	3.30	3.29	3.29
Avg NP_k	20.19	20.21	20.32	20.37	20.51	20.32	20.28	20.31

The average minutes discrepancy per player is around 3 minutes per game. This is not as low as we would ideally like it to be. It is worth noting that the play-by-play's timestamps are in MM:SS format, however, the boxscores are only in MM format. Thus, even a play-by-play with no logging errors will have some discrepancy in player's minutes obtained from the log compared to minutes obtained from the boxscore due to this rounding error. As well, from manually comparing the play-by-play with the video film, there were instances of substoppages that were missed. When this occurs, our agent passes over a point in time in the play-by-play where substitutions actually occurred, as it is not being able to perceive that a substoppage occurred. This limitation is discussed in Section 5.2.1, and possible methods to overcome this limitation is discussed in Section 5.3.1.1. Finally, it is also worth noting that the boxscore minutes recorded are not necessarily true. The likelihood for error in the boxscore minutes exists, given that it is also recorded by humans. However, for the purposes of objectively evaluating our play-by-play algorithm, it is useful to consider it as truth.

The algorithm shows consistent performance across seasons with a very small sample of training data (relative to the number of games that have occurred).

Chapter 5

Discussion

5.1 Summarizing Remarks

In this project, we explored the effectiveness of an automated single agent framework that can clean play-by-play showing a variety of inconsistencies in recorded substitutions. The solution may improve with more data when it is given examples of play-by-play with annotations on which substitutions were recorded correctly. To the best of our knowledge, this is the first type of automated solution that solves the problems that result from human recorded basketball play-by-play. We define two performance measures and show that the agent, with a small amount of initial training data and simple heuristic functions, is objectively successful - with the average absolute difference between minutes extracted from the play-by-play and from the boxscore being approximately three minutes per player. For our specific application, the U Sports league, the analysis that can be derived from the cleaned play-by-play provides access to historical and current statistics beyond the boxscore, such as adjusted plus-minus and WOWY, to coaches and avid fans. For coaching staff, these metrics can inform strategy, decision making and roster management in a similar fashion to how counterparts in the NBA community took advantage of play-by-play derived metrics in the past decade. For media and fans, it introduces and amplifies the growing analytical discourse that our game is seeing. As a league that recently rebranded in 2016 to appeal to a wider audience, as well as to spread stories of young Canadian university athletes (Shoalts, 2016), this is a cost-effective method that can help accomplish both of its stated goals.

This approach can be extended to other lower revenue leagues which suffer the same problems of possessing play-by-play that can contain manual errors which can dramatically affect the results of non-traditional metric calculations; such leagues include the Canadian Collegiate Athletic Association (CCAA) and the Euroleague. Play-by-play is an important data medium, particularly for leagues that cannot afford the infrastructure for video tracking data. We believe this work is an important step in raising the awareness and the standard of analytics for many basketball leagues around the world.

5.2 Limitations

5.2.1 Games With No Substitutions Recorded

The effectiveness of our agent is maximized when it is aware of all the true substoppages that occurred and exactly when they occurred. We mentioned that there are games exhibiting error type 1, that is, with no substitutions recorded whatsoever. Since the frequency of games that exhibit this error is at most one in recent years, this is not a problem of practical importance. However, because the agent only performs substitution imputations at substoppages it is aware of, if a game has no substitutions, then the effectiveness of our agent is severely lowered.

It is possible to clean games that do not have substitutions at all, by considering our set of substoppages as all stoppages that are possible to infer from the play-by-play (turnovers not forced by steal, fouls, timeouts, beginning/end of period). However, this would clearly be a subset of the true substoppages, as there are events in the game where stoppage in play occurs (which would allow a substitution to occur) that are not recorded in the play-by-play (for example, deflections made by the defending team where the ball goes out of bounds and team possession of the ball does not change).

For discussion on potential further work involving inferring latent substoppages, which could address this limitation, please refer to Section 5.3.1.1.

5.2.2 Limited Video Access For Games

Compiling the initial training data for the classification model required video of games. Unfortunately, not every game has a video recording after the fact, and access to video is restricted. We obtained a limited sample of video from coaching staff, but this is not a source of data that is reliably accessible. We recognize that it would be ideal to have more annotated substitutions so that the training set could be larger.

5.2.3 Separate Models For Each Substitution Type

We trained two separate models for substitution correctness classification; one for “enters the game” type substitutions, and one for “goes to the bench” type substitutions. We recognize that while some of the predictors’ influence in the model needed separate coefficients, there are predictors whose coefficients should be equal across both models; that is, the weight of the variable in the classification probability should be independent of the type of the substitution to be classified. For example, looking at Tables 3.2 and 3.3, one could argue that the effect of the feature $X_{6,i}$ (whether the player is recorded in the same substoppage more than once) should be independent of the substitution type and should not have a considerable difference in magnitude between the two logistic regression models.

5.3 Further Work

5.3.1 Customer Lifetime Value

For brevity, we will focus our attention on Customer Lifetime Value (CLV) research that estimates the probability of a customer's latent "death" (death meaning cancellation of business) at any given time after their "birth" (when they began as a customer, such as the first purchase of a product). It is common that a customer does not explicitly tell the business that they have ceased being an active customer when they decide to move on. Customer transaction data can be used to find the time when this probability of death is the highest, using information such as: how long they have been "alive", the total frequency of transactions, and the time of the most recent transaction.

One of the earliest significant contributors to this problem of identifying customers who no longer "alive" (Schmittlein et al., 1987) developed a probability model based on when the last transaction occurred and how many transactions were made in a specified time period, requiring only a few assumptions that are detailed in the paper. Complexity of the model and demanding computational requirements prevented the method from widespread adoption (Fader et al., 2005). More recently, work has been published that improves upon this approach in terms of ease of implementation while remaining competitive in terms of performance (Fader et al., 2005), with only minor adjustments to assumptions from the work it builds upon. Open source software is available that enables relatively easy access to these models and analyses (Pilon, 2017).

5.3.1.1 Estimating Missing Substoppages

As mentioned in the previous section, Section 5.2, our agent is only able to impute missing substitutions at substoppages that are explicitly marked from the play-by-play. If a substitution occurs in the real game, and the scorekeeper does not record any substitutions at all, then it is impossible to know that there was a stoppage in play (at least in the U Sports play-by-play). Thus, it is impossible for our agent to know that it should attempt an imputation. It would be an interesting problem to try and estimate when these latent substoppages occur in the play-by-play.

One approach could be to apply theory from the aforementioned CLV literature (Fader et al., 2005) by creating an analogue between the canonical customer example and our players in the play-by-play. Necessary assumptions for the models require that customers interact while they are "alive" and that there is a probability that they will "die" after some period of time after the initial transaction. In our application with the play-by-play, the "customers" are the players, the "transactions" are the active plays made in the play-by-play log, and the "death" can be defined as when a player actually goes to the bench. Similar to how the time of a customer's true death is unknown, the time at which players go to the bench is unknown to us. When this probability is sufficiently high for a player (or group of players) on the court, this might suggest that a substoppage has occurred.

5.3.1.2 Inferring "goes to the bench" Substitutions

As previously discussed in Section 3.3, inferring which players are on using an activity heuristic is much more straightforward of a task than inferring which players on the court have gone to the bench.

The current implementation imputes "goes to the bench" substitutions using simple logic: at each substoppage, the players on the court are estimated based on an activity heuristic. The players who are imputed to be exiting the game are those such players who were on the court but are not estimated to be on the court any longer.

Instead of using the activity heuristic to decide which players are substituted off by process of elimination of who we now estimate to be on the court, we could use the probabilities derived from the CLV methods to more directly estimate the players who should most likely be going to the bench. More interesting than trying this method would be comparing how the two approaches differ with respect to the performance measures.

5.3.2 Accounting for Scorekeeper Bias

It is possible that there is a scorekeeper effect; for example, some schools may be better at training their scorekeepers than others. The current dataset does not have enough games manually labelled to try and model this. However, with a much larger dataset, it would be a good idea to add a feature that captures the home team (assuming that each team uses the same scorekeepers for every home game).

5.3.3 Other Optimization Approaches

With the problem domain modelled in terms of defining state and utility of the state (with respect to a goal state), there are other optimization techniques that can be used to find a solution state. Two of which I will briefly discuss are genetic algorithms and simulated annealing.

5.3.3.1 Genetic Algorithms

Genetic algorithms (GA) are inspired by the process of natural selection. Out of a population of candidate solutions, some are better than others (or in the evolutionary biology terminology, more "fit" than others). Much like in natural selection, the survival of each individual in the population is based on their fitness. Individuals can randomly mutate (where mutations can either hurt or benefit their fitness) and individuals who survive an iteration breed new offspring that are inserted into the population. Across a sufficient number of generations, the simple combination of breeding and survival result in a set of genetically diverse individuals who are the “fittest”.

These concepts can be applied very successfully in problems that amount to searching in a search space of possible solutions (Mitchell, 1999).

To solve the substitution problem using GA, one would need to work out the details on the breeding operation (more specifically, how two “fit” solutions can be combined to produce a better one).

5.3.3.2 Simulated Annealing

Simulated annealing (SA) simulates the annealing process of metals, which is the physical process of heating a metal and slowly cooling it, over time, to minimize the system energy. If the cooling schedule is sufficiently slow, then the final configuration results in a solid with superior structural integrity (Henderson et al., 2003). SA requires state and the search space to be defined, with an energy function to measure how “good” a state is (one of, or a combination of, our performance measures would make an appropriate energy function).

One could search over time across states, always moving to states that have a lower energy. This search logic is susceptible to being stuck in local minima, as it would never discover a global minimum that requires first traversing to a state with higher energy. The key feature of SA is its acceptance criteria at each iteration: it will always accept a better solution, but there is a probability that the system will move to an objectively worse state with the hopes of finding a global minimum. This probability is a function of the decreasing temperature/time; the probability will be the highest at the beginning of the cooling schedule as the system tries to find the global optimum, and it will be the lowest toward the end of the cooling schedule as the system settles near the optimum.

To solve the substitution problem using SA, one would need to work out the details of what the transition operation should be to move from one state to another.

Bibliography

Fader, P. S., Hardie, B. G. S., and Lee, K. K. (2005). “Counting your customers” the easy way: An alternative to the pareto/nbd model. *Marketing Science, 24(2):275-284.*

Henderson, D., Jacobson, S. H., and Johnson, A. W. (2003). The theory and practice of simulated annealing. *Operations Research and Management Science, Vol 57:287-319.*

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R.* Springer Texts in Statistics.

Mitchell, M. (1999). *An Introduction to Genetic Algorithms.* The MIT Press.

MITSSAC (2017). About. `http://www.sloansportsconference.com/about/`. Accessed: 2017-04-25.

Pilon, C. D. (2017). Lifetime value in python. `https://github.com/CamDavidsonPilon/lifetimes`.

Rudowsky, I. (2004). Intelligent agents. *Americas Conference on Information Systems (AMCIS).*

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach (3rd Edition).* Pearson.

Schmittlein, D. C., Morrison, D. G., and Colombo, R. (1987). Counting your customers: Who-are they and what will they do next? *Management Science, 33(1):1-24.*

Shoalts, D. (2016). CIS rebrands as U Sports, aims to bring student stories to Canadians. `http://www.theglobeandmail.com/sports/canadian-interuniversity-sport-rebrands-as-u-sports/article32452537/`. Accessed: 2017-04-21.

Sill, J. (2010). Improved nba adjusted +/- using regularization and out-of-sample testing. *MIT Sloan Sports Analytics Conference (MITSSAC).*

Sports, U. (2017). Past champions. `http://en.usports.ca/championships/mbkb/past_champs`. Accessed: 2017-04-21.

Witz, B. (2014). College basketball data aplenty for those who can afford it. `https://www.nytimes.com/2014/03/25/sports/ncaabasketball/sportvu-offers-college-basketball-data-for-those-who-can-afford-it.html/`. Accessed: 2017-04-25.