Predictive Hockey Analytics

Erik Hannesson

February 2020

Abstract

It's pretty straightforward, really. We used the National Hockey League's (NHL) API to steal whatever data we could get our grimyc hands on, and then:

Linear Algebra.[1]

And that's about all you need to know.

1 Motivation and Overview

2 Data

So I'm pretty sure we already explained this in the abstract, but I guess we'll vomit it back up here.

The NHL API provides both historical data and live feed data for games currently in play. This allows for incredible diversity in approach and application.

There are two main categories of data that can be analyzed: team data and player data. These can be approached independently, or simultaneously.

Here is a rundown of the main features we use.

Team Statistics

Statistic	Abbreviation
Goals For	GF
Goals Against	GA
Power Play Goals	PPG
Short Handed Goals	SHG
Power Play Percentage	PPP
Penalty Kill Percentage	PK
Blocked Shots	BLK
Shots on Goal	SOG

Skater Statistics

Statistic	Abbreviation
Goals For	GF
Goals Against	GA
Power Play Goals	PPG
Short Handed Goals	SHG
Power Play Percentage	PPP
Penalty Kill Percentage	PK
Blocked Shots	BLK
Shots on Goal	SOG

[1] See Figure A for a helpful infographic.

2.1 Historical Data

2.2 Live Feed

3 Methodology

3.1 Previous Research and Efforts

Uhmm...regression????

Also, apparently game classification accuracy is currently (?) capped out at only $\tilde{6}$6% (look at link on iPad...). Surely we can beat this, no?

3.2 Basic Statistical Analysis

Uhmm, it's probably pretty important to have a section analyzing some "basic" statistical properties before liberally applying machine learning.

3.2.1 Team and Player Variance

We try to create a metric for measuring the variance in both team and individual player performance. This is an incredibly helpful statistic, as it can be a hugely influential factor in the models that we build.

3.3 State Estimation

3.3.1 Overview of Various States

This is where we explain what we mean by "state" in the context of hockey. This includes fully observable states, quasi-observable states (quasi-latent??), and latent states.

One example would be team build or style of play. Some teams are built to be quick on their feet while others are designed to play physically, with the goal of wearing out the other team. We would consider this a quasi-observable state, since you either use ML techniques to determine these states entirely autonomously (unsupervised) or semi-autonomously (semi-supervised), or manually label them based on personal familiarity.[2]

Fully Observable States Description of the fully observable states we deal with.

Latent (Hidden) States Description of the latent (and quasi-latent) states we deal with.

3.3.2 Hidden Markov Models

3.3.3 Mixture Models (Does this go here?)

3.3.4 Kalman Filter (Is this useful...?)

3.4 Time-Series Forecasting

3.4.1 ARIMA Modeling

Team Level

(i) General performance - wins, goals for, etc.

(ii) Specific team performance metrics - PPP, PK, PIM, etc. - to use in classification models for game outcomes *as if* the game already happened (does this make sense?)

[2] Or by whether or not a team's name sounds fierce. However, since there is a lack of academic research concerning such methods, it is hard to say how effective it would be.

Individual Level (and line level??)

Not particularly different from the team level, just for individuals. Obviously, this will encompass different statistics, and therefore different analysis. Also, there is quite a bit more that *could* be done here compared with the team level, I just can't really remember what at the moment.

3.5 Classification Models

3.5.1 GDA? Naive Bayes? What seems *actually* good?

3.6 Dimensionality Reduction

There are a billion and one different statistics that are tracked. Figuring out what is important and what is not is pretty important (and nontrivial). Dimensionality reduction can significantly improve signal-to-noise ratios, etc.

3.6.1 Principal Component Analysis

This will probably be the most useful; I'm not sure if we really need something like NMF...

3.6.2 Nonnegative Matrix Factorization

This *might* be useful. It is currently here mostly as a reminder to think about it.

3.6.3 Randomized Dimensionality Reduction

Yeah just read what's under the nonnegative matrix factorization section right now...

3.7 Causality??

3.7.1 Causal Diagrams

This actually might be a very useful thing to introduce near the beginning, or just use them throughout (causal and just regular DGMs) to help explain the processes attempting to be modeled. Particularly in the HMM and Mixture Model sections (obviously)...

3.8 Reinforcement Learning

Could we run reinforcement learning on historical data but as if it were a live stream??? That is, could we train a model that would predict stuff about a game as its unfolding by using the very large amount of historical data available....?????? That would be awesome.

4 Analysis

5 Conclusion

A Appendix

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Figure 1: For the less technical readers, this is all you really need to know about machine learning and artificial intelligence. (Source: https://xkcd.com/1838/)