Valuing Actions and Ranking Hockey Players With Machine Learning (Extended Abstract)

Oliver Schulte[1]

School of Computing Science, Simon Fraser University,
Burnaby, Canada
oschulte@sfu.ca

Abstract. A fundamental goal of sports analytics is to rank player performance. A common approach is to assign a value to each player action and rank a player by their aggregate action value. A recent AI-based approach is to measure the value of a player's action by how much it increases their team's chance of success, that is, their team's chance of scoring the next goal. This requires a model that outputs a success probability estimate, given a match context and an action. This note describes machine learning techniques for building success probability models from sports data. The most advanced model incorporates *playing style representations* for 1K+ NHL players. The resulting action values, player rankings, and player rankings are illustrated with data from the National Hockey League.

Keywords: Sports Analytics · Machine Learning · Ice Hockey · Success Probabilities · Agent Representation.

1 Introduction: Success Probabilities in Sports Analytics

During a match, each action by a sports team is directed towards maximizing the chance of future success. Therefore the *probability of (future) success* is a key statistical quantity for evaluating the strength of a team, the impact of an action, and the contributions of a player. This note accompanies a talk that shows how success probabilities can be used to value actions and ranking players, and describes techniques for estimating them from data. The different strengths and weaknesses of an individual player can be measured by comparing how they impact their team's of success compared to other players carrying out the same action in the same context. Technically, we implement this comparison by learning a context-aware *player representation* for each player that captures their impact in a given match context.

2 Defining Success Probabilities

For the purposes of this note, success is flexibly defined as a binary event that a team seeks to bring about. An analyst is free to define success in different ways depending on the question they are investigating. Examples of success concepts that have appeared in the literature include the following.

– Winning the Match [8].
– Scoring a goal within a short time interval (e.g. 1 minutes) [11].
– Scoring a goal within the next 5 actions [2].
– Scoring the next goal in the match [10].
– Drawing the next penalty [10]. This is a failure event that would be interesting to a coach who is concerned to minimize the number of penalties incurred by their team.

The notation

$$P(s_i|\mathbf{x}_t)$$

denotes the probability that team i achieves (future) success given the **current match context** $\mathbf{x}_t$. A success probability is a dynamic quantity; a success probability ticker shows an estimated probability for each time in a match [5]; see Figure 1.

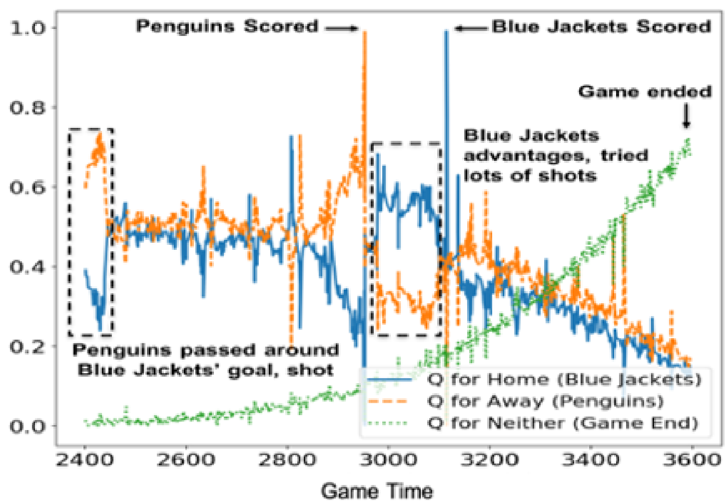

Fig. 1. A success probability ticker for a match between the Penguins and the Blue Jacket. The y-axis shows the estimated probability of scoring the next goal.

3 From Success Probabilities to Action Values

Given success probabilities, we can assign a value to actions called the **impact** of an action occurring at time $t + 1$ [5].

$$impact_i(t + 1) \equiv P(s_i|\mathbf{x}_{t+1}) - P(s_i|\mathbf{x}_t)$$

Thus the impact of an action is the difference in success probabilities before and after the action occurred. Figure 3 shows boxplots for impact values. Note that impact values can vary widely for the same action, depending on context.

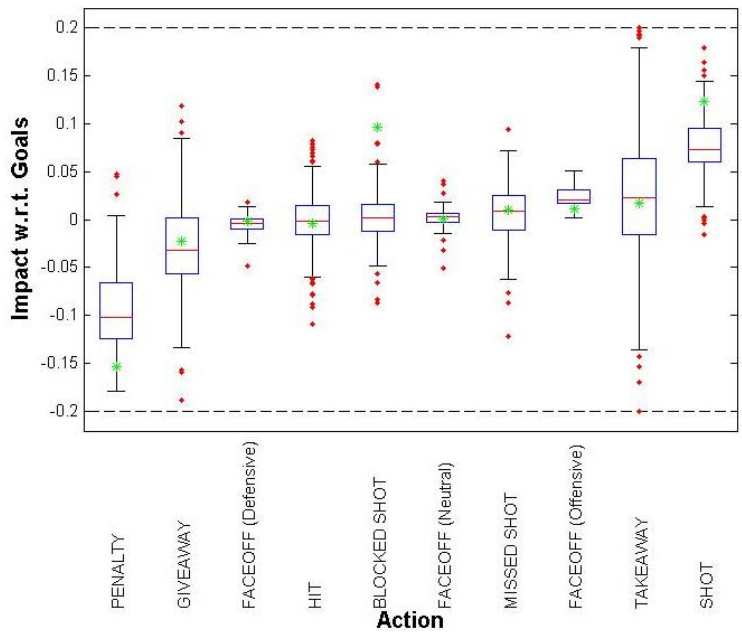

Fig. 2. Impact on the probability of Scoring the Next Goal. Higher numbers are *better* for the team that performs the action. Action Impact Values vary with context. The central mark is the median, the edges of the box are the 25th and 75th percentiles. The whiskers are at the default value, approximately 2.7 s.d. Based on the model of [10].

4 From Action Values to Player Ranking

We can compute a player performance metric from impact values in a straightforward way: for each player, and for each of their actions, we can compute the impact of the action. The **goal impact metric** (GIM) is simply the total impact of the player's actions; see Table 1.

Our metric can be used to *identify undervalued players.* For instance, Johnny Gaudreau and Mark Scheifele drew salaries below what their GIM rank would suggest. Later they received a $\$5M+$ contract for the 2016-17 season.

While we do not have ground truth for player rankings, the goal impact player rankings have been validated indirectly in several ways [5,9,8], and in several sports in addition to hockey, such as soccer [2,4,3] and basketball [1].

Table 1. 2015-2016 Top-10 Player Impact Scores. Based on the model of [5].

Name	Impact	Assists	Goals	Points	+/-	Salary
Taylor Hall	96.40	39	26	65	-4	$6,000,000
Joe Pavelski	94.56	40	38	78	25	$6,000,000
Johnny Gaudreau	94.51	48	30	78	4	$925,000
Anze Kopitar	94.10	49	25	74	34	$7,700,000
Erik Karlsson	92.41	66	16	82	-2	$7,000,000
Patrice Bergeron	92.06	36	32	68	12	$8,750,000
Mark Scheifele	90.67	32	29	61	16	$832,500
Sidney Crosby	90.21	49	36	85	19	$12,000,000
Claude Giroux	89.64	45	22	67	-8	$9,000,000
Dustin Byfuglien	89.46	34	19	53	4	$6,000,000

5 Learning Success Probability Models

[12] gives a brief overview of success probability learning from event data, also known as play-by-play data, such as that shown in Table 2.

Table 2. Sample Play-By-Play Data in Tabular Format.

gameId	playerId	period	teamId	xCoord	yCoord	Manpower	Action Type
849	402	1	15	-9.5	1.5	even	lpr
849	402	1	15	-24.5	-17	even	carry
849	417	1	16	-75.5	-21.5	even	check
849	402	1	15	-79	-19.5	even	puckprot.
849	413	1	16	-92	-32.5	even	lpr
849	413	1	16	-92	-32.5	even	pass
849	389	1	15	-70	42	even	block
849	389	1	15	-70	42	even	lpr
849	389	1	15	-70	42	even	pass
849	425	1	16	-91	34	even	block
849	395	1	15	-97	23.5	even	reception

A straightforward approach is to annotate each event at time t with a binary target $y_{i,t} \in \{0, 1\}$ that denotes whether team i acting at time t achieved future success after time t. For example in the game of Figure 1, the Penguins scored around time $t = 3,900$ sec. So for all previous times $2,400 < t' < 3,900$, we would have $y_{Penguins,t'} = 1$ and $y_{Flyers,t'} = 0$. Then estimating success probabilities can be modelled as predicting a binary label given the information $\mathbf{x}_t$ available at time t, for example with standard classification packages like R, Weka, scikit-learn.

Applying classification packages to pairs $\langle \mathbf{x}_t, y_{i,t} \rangle$ implicitly treats future success at different times as independent, which clearly is not correct. An alternative are **reinforcement learning** methods [14] such as temporal difference learning, which use a training objective that enforces temporal consistency between different predictions. For a brief overview of success probability learning methods see [12].

6 Generative Model for Learning Player Representations

In current deep learning models, discrete entities are represented by real-valued vectors called *representations* or *embeddings*. For example, the famous large language models map each word to an embedding that depends on the context of a word in a sentence. Computing a representation of an entity is similar to assigning a type to the entity, except that the entity is assigned an embedding in a real-valued space, rather than a discrete type. Following the generative approach of [6], we build a generative model for player identities. The high-level idea is that the model uses player embeddings to generate, for each match context and action, which player is likely to have performed the action. Thus the model answers an *identification* question like "who took the shot 30 minutes into the game?" [7].

A formal specification of our generative model is as follows. A **dynamic player representation** is a time-dependent vector $\mathbf{z}_t \in \mathbb{R}^d$ where d is the embedding dimension. Player generation utilizes two components as follows.

1. A **context prior** $p(\mathbf{z}_t|\mathbf{x}_t, a_t)$. Given a context vector $\mathbf{x}_t$ summarizing the match context at time t and a current action a_t, we sample an embedding from the context prior.
2. A **decoder** model $p(pl_t|\mathbf{z}_t)$. Given an embedding, we sample a player from the decoder distribution.

The context prior and decoder can each be implemented by a neural network. For a training objective, we use the **ELBO** approximation for the generative model log-likelihood [13]:

$$\ln p(pl_t|\mathbf{x}_t, a_t) = \ln \int_{\mathbf{z}_t} p(pl_t|\mathbf{z}_t) p(\mathbf{z}_t|\mathbf{x}_t, a_t) \geq$$
$$E_{\mathbf{z}_t \sim q(\mathbf{z}_t|pl_t, \mathbf{x}_t, a_t)} \ln p(pl_t|\mathbf{z}_t) - KL(q(\mathbf{z}_t|pl_t, \mathbf{x}_t, a_t)||p(\mathbf{z}_t|\mathbf{x}_t, a_t)) \quad (1)$$

where q is the **encoder** model (approximate posterior) that takes as input the current player, match context, action, and outputs a distribution over match embeddings (e.g., the parameters of a Gaussian distribution). Since the right-hand side (1) is a lower bound, maximizing it with respect to the encoder, decoder, and prior models will drive training towards a maximum of the model log-likelihood. The KL divergence acts as a regularizer on embeddings. It induces a *shrinkage effect* that helps the model generate embeddings for new players with few observations. *After training we can use the encoder to generate context-aware player embeddings.*

Figure 3 visualizes the resulting embeddings in 2D for 1003 NHL players across a number of match contexts. Embeddings for the same player position are similar indicating that the model matches known player types. Embeddings for a single player are similar indicating that the model captures a consistent playing style. Embeddings for similar locations are similar indicating that the model captures action context.

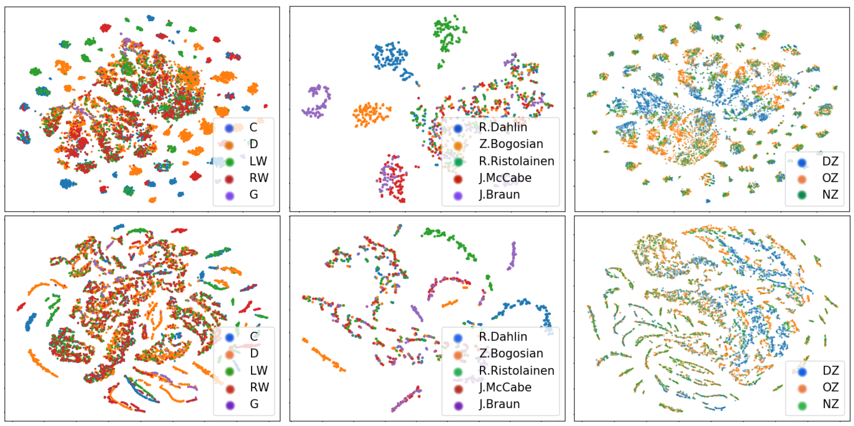

Fig. 3. Embedding visualization. Each data point corresponds to a player embedding conditioning on the game context at the current time step t. The player embeddings are labelled 1) player positions (on the *left* column, including Center (C), Defense (D), Left-Wing (LW) and Right-Wing (RW) and Goalie (G)) 2) 5 selected defence men (on the *middle* column) and 3) player locations (on the *right* column, including Defence Zone (DZ), Neutral Zone (NZ) and Offensive Zone (OZ)).

Adding the learned player embeddings to the context vector $\mathbf{x}_t$ improves predictive performance on tasks such as predicting the success of a shot and the final game outcome; for detailed experimental results please see [6].

7 Conclusion

Estimating success probabilities is a basic statistical problem in hockey analytics. A good success probability model can be leveraged to solve important analytics problems such as quantifying the value of an action and the contributions of a player. Machine learning models can include rich match contexts to provide useful success probabilities. Including a representation of the currently acting player in the match context improves the complexity and accuracy of a hockey model. We showed how player representations can be learned by generative AI methods.

References

1. Cervone, D., D'Amour, A., Bornn, L., Goldsberry, K.: Pointwise: Predicting points and valuing decisions in real time with NBA optical tracking data. In: 8th Annual MIT Sloan Sports Analytics Conference, February. vol. 28 (2014)
2. Decroos, T., Bransen, L., Haaren, J.V., Davis, J.: Actions speak louder than goals: Valuing player actions in soccer. In: Proceedings of the 25th International Conference on Knowledge Discovery & Data Mining (KDD-19). pp. 1851–1861 (2019)
3. Fernández, J., Bornn, L., Cervone, D.: A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions. Machine Learning **110**(6), 1389–1427 (2021)
4. Liu, G., Luo, Y., Schulte, O., Kharrat, T.: Deep soccer analytics: learning an action-value function for evaluating soccer players. Data Min. Knowl. Discov. **34**(5), 1531–1559 (2020). https://doi.org/10.1007/s10618-020-00705-9, https://doi.org/10.1007/s10618-020-00705-9
5. Liu, G., Schulte, O.: Deep reinforcement learning in ice hockey for context-aware player evaluation. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. pp. 3442–3448. International Joint Conferences on Artificial Intelligence Organization (7 2018). https://doi.org/10.24963/ijcai.2018/478, https://doi.org/10.24963/ijcai.2018/478
6. Liu, G., Schulte, O., Poupart, P., Rudd, M., Javan, M.: Learning agent representations for ice hockey. In: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2020, December 6-12 2020 (2020)
7. Lu, W.L., Ting, J.A., Little, J.J., Murphy, K.P.: Learning to track and identify players from broadcast sports videos. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013), https://www.cs.ubc.ca/~murphyk/papers/weilwun-pami12.pdf
8. Pettigrew, S.: Assessing the offensive productivity of nhl players using in-game win probabilities. In: 9th Annual MIT Sloan Sports Analytics Conference (2015)
9. Routley, K.: A Markov Game Model for Valuing Player Actions in Ice Hockey. Master's thesis, Simon Fraser University (April 2015)
10. Routley, K., Schulte, O.: A markov game model for valuing player actions in ice hockey. In: Uncertainty in Artificial Intelligence (UAI). pp. 782–791 (2015)
11. Schuckers, M., Curro, J.: Total hockey rating (thor): A comprehensive statistical rating of national hockey league forwards and defensemen based upon all on-ice events. In: MIT sloan Sports Analytics Conference (2013)
12. Schulte, O.: Valuing actions and ranking hockey players with machine learning. In: Lambrix, P., Carlsson, N., Vernblom, M. (eds.) Linköping Hockey Analytics Conference LINHAC. pp. 2–9. Linköping Electronic Conference Proceedings 191 (2022)
13. Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. Advances in neural information processing systems **28** (2015)
14. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Introduction to Statistical Relational Learning, chap. 4, pp. 93–127. MIT Press (2007)