Learning a Markov Model for Evaluating Soccer Decision Making

Maaike Van Roy [1 2] Pieter Robberechts [1 2] Wen-Chi Yang [1 2] Luc De Raedt [1 2] Jesse Davis [1 2]

Abstract

Reinforcement learning techniques are often used to model and analyze the behavior of sports teams and players. However, learning these models from observed data is challenging. The data is very sparse and does not include the intended end location of actions which are needed to model decision making. Evaluating the learned models is also extremely difficult as no ground truth is available. In this work, we propose an approach that addresses these challenges when learning a Markov model of professional soccer matches from event stream data. We apply a combination of predictive modeling and domain knowledge to obtain the intended end locations of actions and learn the transition model using a Bayesian approach to resolve sparsity issues. We provide intermediate evaluations as well as an approach to evaluate the final model. Finally, we show the model's usefulness in practice for both evaluating and rating players' decision making using data from the 17/18 and 18/19 English Premier League seasons.

1. Introduction

Reinforcement learning techniques are increasingly being applied to analyze sports such as ice hockey (Liu & Schulte, 2018; Luo et al., 2020; Routley & Schulte, 2015; Schulte et al., 2017), soccer (Fernández et al., 2021; Hirotsu & Wright, 2002; Liu et al., 2020; Rudd, 2011; Singh, 2019; Van Roy et al., 2021; Yam, 2019), basketball (Cervone et al., 2016; Sandholtz & Bornn, 2018; 2020; Wang et al., 2018), and American football (Goldner, 2012). These models leverage the large amounts of data that are being collected from matches, which include line-up information, specific actions or events that occur, and sometimes even include information about the location of all players at each moment of the match. Using this data, reinforcement learning techniques are able to model the observed team and player behavior. Such models have a variety of use cases including performing match analysis (Fernández et al., 2021), aiding tactical planning (Hirotsu & Wright, 2002), evaluating the effect of different strategies (Sandholtz & Bornn, 2018; 2020; Van Roy et al., 2021) and rating the actions of players which is useful for player scouting (Cervone et al., 2016; Liu & Schulte, 2018; Liu et al., 2020; Luo et al., 2020; Routley & Schulte, 2015; Rudd, 2011; Schulte et al., 2017; Singh, 2019; Van Roy et al., 2020; Yam, 2019).

In this paper, we will focus on learning a Markov model of professional soccer matches with the goal of aiding in-game decision making. The model will be learned on the basis of event stream data, which is a type of data that annotates various events such as passes, tackles, interceptions, and shots that occur during a match. Moreover, it also records information like the location of the event on the pitch, the players involved in the event, and the time the event occurred. Working with such data poses a number of challenges from a learning perspective. First, the observational nature of the data means that when an action is unsuccessful, its intended end location is not recorded and hence unknown. For example, if a player attempts a cross that the opposition clears, we are unsure of where the player was aiming. Second, the data is sparse. A season is relatively short and the dynamic nature of the game makes it such that teams rarely perform the exact same action multiple times. Finally, evaluating and validating the learned models is extremely difficult. For other tasks like rating individual actions, ground truth ratings simply do not exist. For tasks like assessing changes to in-game decision making, validation becomes even harder as the proposed changes cannot be implemented in practice solely for the sake of evaluation.

Our work uses a Markov Decision Process (MDP) to model soccer and addresses each of the aforementioned challenges when learning the model from event stream data. First, we learn a predictive model to predict the intended end locations of players' actions using a combination of domain knowledge and predictive modeling. Second, we employ a hierarchical Bayesian approach to learn the transition model which helps mitigate sparsity issues by using a prior based on a "typical team". The model is then specialized to an individual team on the basis of their data. Third, we evaluate each step and outline an approach to validate the final

[1]Department of Computer Science, KU Leuven, Leuven, Belgium [2]Leuven.AI. Correspondence to: Maaike Van Roy <maaike.vanroy@kuleuven.be>.

Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 38^{th} International Conference on Machine Learning, 2021.

model. Finally, we show how our framework can be used for both evaluating and rating players via three illustrative use cases using data from the 17/18 and 18/19 English Premier League seasons. The first two use cases focus on *reasoning about the learned MDP* to gain insight into shooting behavior from outside the penalty box. The prevalence of these "long-distance" shots has declined in recent seasons, which has been driven by the analytical insight that fewer high quality shots will yield more goals than taking many low quality shots.[1] Hence, teams are forgoing these long shots in the hopes of generating a better shot down the line. One case reasons about the model to investigate whether directly shooting or moving once prior to shooting is more likely to lead to a goal when possessing the ball outside the penalty box. The results indicate that team-specific locations exist where directly shooting is more advantageous than moving once prior to shooting. The following case reasons about the effect of modifying a team's observed long-distance shooting behavior. The results contradict current wisdom and indicate that teams should shoot more often from distance. Specifically, shooting more often from the long-distance locations identified in the first use case would result in teams scoring up to three more goals per season. The final use case illustrates how the riskiness of each player's decision making can be rated, which could be used for scouting. We provide and discuss rankings of the least and most risky players.

2. Related work

In the literature, addressing issues regarding sparsity in sports models is done in various ways. While including fine-grained location and context information would allow for a more detailed analysis, it also increases the sparsity of the data. Therefore, most approaches trade off usefulness and sparsity by applying a hand-crafted grid over the field and only sometimes include context such as players (Goldner, 2012; Luo et al., 2020; Routley & Schulte, 2015; Rudd, 2011; Schulte et al., 2017; Singh, 2019; Van Roy et al., 2021; Yam, 2019). Recent work in basketball goes one step further and applies Bayesian approaches on top of a fine-grained and contextualized state space to estimate the transition probabilities and thereby mitigates any remaining sparsity issues (Cervone et al., 2016; Sandholtz & Bornn, 2018; 2020). This approach has also been proposed in soccer, but without any validation on real-life data and with a rudimentary model containing only four states (Hirotsu & Wright, 2002). To the best of our knowledge, these approaches have not been applied to a more fine-grained Markov model for soccer in order to resolve sparsity issues.

Evaluation and validation of these models is often not straightforward and the approach taken typically depends on the use case. In the case of valuing player actions, some approaches validate their ratings against other metrics such as salaries, goals, and assists (Liu & Schulte, 2018; Liu et al., 2020; Luo et al., 2020; Routley & Schulte, 2015; Schulte et al., 2017). While correlation with these metrics might indicate correctness, a perfect correlation indicates that the proposed approaches do not provide any new insights. As there are no ground truth labels available, it is hard to quantitatively validate the correctness and usefulness of these metrics and one often resorts to domain knowledge. In the case of validating the learned playing style, validation against the team's actual playing style is equally challenging and mainly done with visual inspection or against domain knowledge (Schulte et al., 2017; Singh, 2019). Quantitative validation of the complete Markov model is often left out. However, in case of intermediate models (e.g., Bayesian approaches), evaluation of those is often done to yield insight into the correctness of the final model (Cervone et al., 2016; Fernández et al., 2021; Sandholtz & Bornn, 2018; 2020).

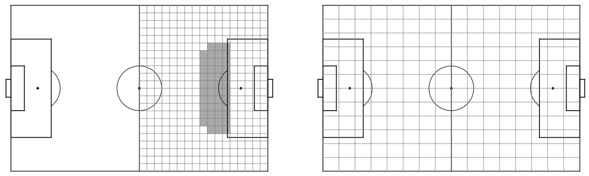
[1] http://thepowerofgoals.blogspot.com/2014/02/twelve-shots-good-two-shots-better.html

3. A Markov model for soccer

We model the in-game behavior of a team possessing the ball using a MDP. This formalism allows modeling and thus analyzing the different in-game decisions players make. When a player is attempting to move the ball, we explicitly consider the *intended* location where the players wants to move the ball (e.g., the end location of a pass). Existing Markov models for soccer have used the *observed* end location of a movement action instead (Rudd, 2011; Singh, 2019; Yam, 2019), which can differ from the *intended* one on failed actions (e.g., an intercepted pass).

Formally, the MDP consists of the following parts: the state space $\mathcal{S}$, the set of actions $\mathcal{A}$, the transition function P, the policy π, the reward function R, and a discount factor γ. Next, we discuss each of these parts.

State space: Defining the state space is intertwined with the type of data that is used. We will use event stream data which is a common source of data that records information about all on-the-ball actions (e.g., pass, cross, dribble, shot) in a soccer match. Per action, various features are recorded such as the player that made the action, the time at which it was played, the start and end locations, whether it was successful or not, etc. However, it does not contain any information about the whereabouts of the other players on the field. Thus, the amount of context that can be included in the model is limited. Given this data set and our aim to analyze in-game behavior, we define the state space entirely based on the location of the ball. Formally, the state space is defined as $\mathcal{S} = \mathcal{L} \cup \mathcal{E}$. Here, $\mathcal{L}$ denotes the transient states which are defined by their location on the pitch, and $\mathcal{E} = \{lost\_possession, no\_goal, goal\}$ denotes the absorb-



(a) Offensive grid (b) Regular grid

Figure 1: Illustration of two sets of transient states that will be used. Each transient state is determined by their location on the field. Teams play left to right. In (a), the defensive half denotes one state, the offensive half is split into zones of 3m × 3m. The shaded area denotes the set of states from which teams shoot when outside the penalty box and will be used in the experiments. In (b), the field is divided into states using a regular grid of 12 × 16 cells.

ing states which signify the different ways in which the ball can be lost to the opponent. With this state space, the MDP describes how a team moves the ball on the field during ball possession. The most appropriate state space depends on the considered use case. When analyzing shooting behavior, a more fine-grained state space around the opponent's goal is needed. When rating players' actions, the entire field must be considered. Therefore, we will use the two different transient state spaces shown in Figure 1 in our experiments.

Action set: For each state $s \in \mathcal{L}$, we consider the same set of actions: $\mathcal{A} = \{shoot, move\_to(s) | s \in \mathcal{L}\}$. Each $move\_to(s)$ action denotes the intended end state of the chosen action. This enables analyzing players' in-game decision making. For example, it allows for assessing whether a movement action in a certain direction is more beneficial than another, which is not easily done with the more common way of modeling the actions.

Transition function: The transition function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ models the success and fail probabilities of each action. For the absorbing states $\mathcal{E}$, $\mathcal{A} = \emptyset$, and the transition function denotes a self-loop with probability one. For each state $s \in \mathcal{L}$, the transition function is defined as follows:

- $P(s, move\_to(s'), s')$ denotes the probability of successfully moving to s' from s;
- $P(s, move\_to(s'), lost\_possession)$ denotes the probability of unsuccessfully moving to s' from s, this is equal to $1 - P(s, move\_to(s'), s')$;
- $P(s, shoot, goal)$ denotes the probability of scoring a goal from s;
- $P(s, shoot, no\_goal)$ denotes the probability of failing to score a goal when shooting from s, this is equal to $1 - P(s, shoot, goal)$;
- $P(s, a, s')$ = 0 in all other cases.

Policy: The policy π defines the probability distribution over actions for each possible state: $\pi(a|s) = Pr[A = a|S = s]$. It denotes how teams choose different actions in each state, and together with the transition function it completely defines the in-game behavior of the team.

Reward function and discount factor: The reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ assigns a reward of one when a goal is scored from a state $s \in \mathcal{L}$, i.e., $R(s, shoot, goal) = 1$, and a reward of zero otherwise. Thus, it mimics the rewards received in a real-life soccer game. The discount factor γ is set to 1, signifying that every goal is equally important.

4. Learning the model from event stream data

Learning the MDP for a team requires estimating the transition function and policy from historical data. However, this is challenging for two reasons. The first arises due to our choice to include the intended end location of actions in our model. A consequence of this is that estimating the transition probabilities is not straightforward because it requires knowing both the number of successful and *unsuccessful attempts* to move the ball from any zone $s \in \mathcal{L}$ to any other zone $s' \in \mathcal{L}$. Unfortunately, for failed (i.e., unsuccessful) moves, the *intended* end location, and thus the number of unsuccessful moves, is unknown. For example, consider an intercepted pass. Event stream data records the location where the ball was intercepted but not the location it was intended to reach. Hence, the number of unsuccessful moves cannot be directly estimated from the data. We solve this problem by using a combination of predictive modeling and domain knowledge to estimate a distribution over possible intended end locations for each failed action.

The second challenge arises due to the nature of soccer itself. Often, it is most informative to perform a team-level analysis due to variations in style of play and team strength. However, the amount of relevant data available per team can be limited. For example, a typical team will perform between 400 and 700 shots per season. While passes are much more common (e.g., teams perform between 10,000 and 25,000 passes per season), they are not evenly distributed on the pitch. There is much more action in the midfield and relatively little near the opponents' goal. Data more than a few seasons old is of limited use due to turnover in managers and players. Moreover, general game play trends evolve over time. The net effect is that the data can be very sparse in certain regions of the pitch. We overcome this challenge by using a Bayesian approach to estimate the parameters.

We will use event data from the 17/18 and 18/19 English Premier League (EPL) seasons to construct and evaluate our models. This data was scraped from http://www.whoscored.com. We build models for the following six teams that played in the EPL during both sea-

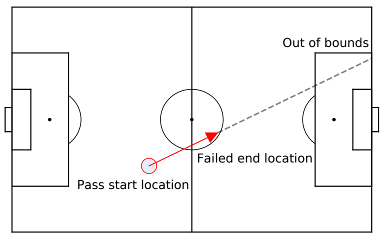


Figure 2: Illustration of a failed pass. Its intended end location most likely lies along the grey line from its start location and passing through its failed end location.

sons: Burnley, Chelsea, Huddersfield Town, Manchester City, Manchester United, and Newcastle United. Next, we describe each aspect of learning the model in more detail.

4.1. Predicting intended end locations

To estimate the number of unsuccessful moves from one zone to another, we need to predict the intended end location for each failed movement action. This will be done on a per-team basis to capture the differences per team. At a high level, the data records three types of actions that entail moving the ball and for which predictions are necessary: passing, crossing and dribbling. We treat each of these separately:

Passes: To predict the intended end location of failed passes, we train a gradient boosted trees ensemble on the successful passes of a team. The features used include the pass' start location, body part used to execute the pass, and direction of the pass, as well as the start and end locations of the previous three actions in the sequence. The locations refer to the discrete transient states $\mathcal{L}$. For each failed pass, we use the ensemble to predict a probability distribution over its intended end location. This distribution can be further improved by post-processing the predictions using domain knowledge. Generally, a pass travels in a straight line. Therefore, its intended end location likely lies along the line (Figure 2) through its observed start and (failed) end location, which is recorded in the data. Moreover, for an *intercepted* pass its intended end location likely lies on this line after the failed end location because interceptions occur *before* a pass reaches its target. Therefore, we post-process this distribution by only retaining locations that fall on this line, and renormalize the probabilities for these states.

Crosses: As with passes, a gradient boosted trees ensemble using the same feature set is trained on successful crosses. For all failed crosses, the ensemble again predicts a probability distribution over all possible intended end locations, which can again be improved with domain knowledge. Crosses are passes that generally originate from the flanks near the opponent's goal and cross the area in front of the goal. We simplify this definition and assume crosses travel through the air on a parabolic path. Therefore, the most likely intended end location will be a location in the near vicinity of the failed one. Hence, we post-process the distribution by only retaining locations within a radius r from the failed end location, and renormalize the probabilities for these states.

Dribbles: A dribble is a local action where a player either tries to carry the ball forward or tries to move past an opposing player. A failed dribble indicates the player lost control of the ball (e.g., was tackled) and the other team recovers possession. Given the extremely local nature of this action, we assume that the intended end location of a failed dribble is simply the observed end location where the dribble failed.

Training the team-specific models to predict the end location of passes and crosses is done using XGBoost.[2] The training data consists of the successful passes or crosses performed by the team. We use a 70-30 train-test split and perform 5-fold cross-validation on the training data to tune XGBoost's hyperparameters. The test set will be used for evaluation. For crosses, the radius $r = 2.5$ meters, which is a reasonable error margin for professional soccer players.

4.2. Learning the transition model from data

With the intended end location of each movement action known, the transition model can be learned from the data. To overcome the challenges of sparse data, we apply a Bayesian hierarchical modeling approach to learn the transition model. The hierarchical approach allows sharing information between teams. Conceptually, this can be viewed as having a generic model for how teams behave that is then adapted based on the specific observed actions that a team performs during matches. If there is strong empirical evidence that a team deviates from a "typical team", the model will pick it up. However, in locations where there is little data for a team, the parameters will shrink towards the prior.

We define separate models to estimate the transition function and policy of each team as follows:

Transition probability model: To learn the probabilities of the transition function for a specific team t, we model the probability that a chosen action $a \in \mathcal{A}$ in a certain state $s \in \mathcal{L}$ succeeds as a Bernoulli random variable $O_{t,s,a}$:

$$\begin{aligned} O_{t,s,a} &\sim \text{Bernoulli}(p_{t,s,a}) \\ p_{t,s,a} &= \text{invlogit}(\gamma_{t,s,a}) \\ \gamma_{t,s,a} &\sim \mathcal{N}(\mu_{s,a}, \sigma^2_{t,s,a}) \\ \sigma^2_{t,s,a} &\sim \text{Half-Normal}(5.0) \end{aligned}$$

Here, $\gamma_{t,s,a}$ denotes the log-odds of team t successfully com-

[2] See https://xgboost.ai/. We used XGBoost v1.4.

pleting action a in state s and is normally distributed with an overall prior mean $\mu_{s,a}$ and a team-dependent variance $\sigma^2_{t,s,a}$. The prior mean is computed based on the data of all other teams by means of counting and Gaussian smoothing to ensure spatial coherence. The team-dependent variance is half-normally distributed with a scale factor of 5.0, corresponding to a weakly informative prior.

Policy model: To learn the probabilities of the policy for a specific team t, we model the probability of choosing a specific action in state s as a Categorical random variable $A_{t,s}$ with action type probabilities $\vec{p}_{t,s}$:

$$\begin{aligned} A_{t,s} &\sim \text{Categorical}(\vec{p}_{t,s}) \\ \vec{p}_{t,s} &= \text{softmax}(\vec{\lambda}_{t,s}) \\ \lambda_{t,s,a} &\sim \mathcal{N}(\alpha_{s,a}, \nu^2_{t,s,a}) \\ \nu^2_{t,s,a} &\sim \text{Half-Normal}(5.0) \end{aligned}$$

Here, $\vec{\lambda}_{t,s}$ denotes the log-odds of team t choosing each different action $a \in \mathcal{A}$ in state s. Each $\lambda_{t,s,a}$ is normally distributed with an overall prior mean $\alpha_{s,a}$ and a team-dependent variance $\nu^2_{t,s,a}$ for each chosen action a. Similarly to the transition probability model, the prior mean is computed based on the data of all other teams with simple counting and Gaussian smoothing. The team-dependent variance is half-normally distributed with a scale factor of 5.0, which corresponds to a weakly informative prior.

We model and train each of the above described models for each team using the probabilistic programming package PyMC3.[3] The training of the models is done using PyMC3's Auto-Differentiation Variational Inference (ADVI) implementation, which allows scaling to large data sets. To allow for the use of the predicted end location distributions of Section 4.1, we weight each observation in the data set according to its estimated intended end location. For example, successful examples receive a weight of one as their intended end location is known. Failed examples are repeated n times where n is equal to the number of predicted end locations with a non-zero probability and each example is weighted according to that probability. This ensures that failed examples are weighted equally compared to successful examples. We used 40,000 iterations to fit the policy models and 100,000 iterations to fit the transition probability models until convergence. For each model, we used the average of 4,000 samples to compute the final probabilities.

5. Evaluating the learned models

Evaluating and validating the learned models is not straightforward as there is no ground truth model available. In this section, we will discuss how each aspect of the model can be evaluated and propose a method to validate the final model.

[3]See https://docs.pymc.io/. We used PyMC3 v3.11.

Table 1: AUROC and Brier scores for the end location prediction models averaged over the six teams (O = offensive state space, F = full regular grid). The baseline predicts the distribution over observed end locations in the training set. DK refers the post-processing step using domain knowledge. For AUROC, higher values are better. For the Brier score, lower values are better.

Action	Model	AUROC	Brier Score
Pass (O)	Baseline	0.50 ($\pm$ 0.00)	0.77 ($\pm$ 0.03)
	XGBoost	0.74 ($\pm$ 0.05)	0.60 ($\pm$ 0.04)
	DK	0.99 ($\pm$ 0.00)	0.54 ($\pm$ 0.02)
Pass (F)	Baseline	0.50 ($\pm$ 0.00)	0.99 ($\pm$ 0.00)
	XGBoost	0.88 ($\pm$ 0.03)	0.93 ($\pm$ 0.02)
	DK	0.99 ($\pm$ 0.00)	0.69 ($\pm$ 0.03)
Cross (O)	Baseline	0.50 ($\pm$ 0.00)	0.96 ($\pm$ 0.01)
	XGBoost	0.52 ($\pm$ 0.03)	0.97 ($\pm$ 0.02)
	DK	0.95 ($\pm$ 0.01)	0.77 ($\pm$ 0.06)
Cross (F)	Baseline	0.50 ($\pm$ 0.00)	0.89 ($\pm$ 0.02)
	XGBoost	0.53 ($\pm$ 0.02)	0.89 ($\pm$ 0.01)
	DK	0.95 ($\pm$ 0.02)	0.58 ($\pm$ 0.08)

5.1. Evaluating the end location predictions

To evaluate the performance of the end location prediction models, we use the held out test set of successful actions as defined in Section 4.1. Table 1 reports the area under the ROC curve (AUROC) and the original Brier score (Brier, 1950) averaged across the six teams and for both proposed state spaces. AUROC measures a model's ability to distinguish between classes. The Brier score assesses if the model's probability estimates are well-calibrated. Calibration is arguably more important in this case since we use the estimates to learn the transition model of the MDP. The baseline model corresponds to predicting a prior probability of an action ending in each location. The learned models outperform the baseline and using domain knowledge to post-process the predictions substantially improves performance.

5.2. Evaluating the Bayesian models

To evaluate the predictive accuracy of the Bayesian models, we compute the expected log pointwise predictive density using Pareto-smoothed importance sampling leave-one-out cross-validation (PSIS-LOO-CV).[4] This approach efficiently estimates the expected log pointwise predictive density, alleviating the computationally expensive burden of refitting and evaluating the models for each left out example.

Table 2 shows the estimated log pointwise predictive density values for our proposed Bayesian models, averaged over all teams and for both state spaces. The results are compared to the results of a non-hierarchical approach in which the prior means $\mu_{s,a}$ and $\alpha_{s,a}$ are estimated for each team separately with a prior normal distribution with zero mean and standard

[4]See https://arviz-devs.github.io/arviz/. We used the PSIS-LOO-CV implementation from Arviz v2.0.

Table 2: Expected log pointwise predictive density (in thousands) using PSIS-LOO-CV for each proposed hierarchical model (O = offensive state space, F = full regular grid). Results are averaged ($\pm$ standard error) over all six teams and compared against non-hierarchical versions of the models. Higher values indicate models with better predictive accuracy.

Model	$\pi(\cdot)$	$P(\cdot)$
Unpooled (O)	-182.7 ($\pm$ 0.80)	-28.8 ($\pm$ 0.14)
Hierarchical (O)	-167.6 ($\pm$ 0.79)	-26.1 ($\pm$ 0.18)
Unpooled (F)	-108.1 ($\pm$ 0.47)	-13.7 ($\pm$ 0.08)
Hierarchical (F)	-101.5 ($\pm$ 0.45)	-11.3 ($\pm$ 0.09)

deviation 5.0. The results show that the hierarchical models outperform the non-hierarchical approaches for both the policy and transition probability models and for both state spaces. The ability to use information about a global prior helps with resolving sparsity issues and in turn increases predictive accuracy.

5.3. Evaluating the complete Markov model

Finally, as an additional check, the correctness of the complete Markov model can be evaluated by comparing statistics derived from the model to those derived from the raw data. Specifically, we compute the value of each state, both empirically and with the value function (Bellman, 1966):

$$V_\pi(s)=\sum_{a\in\mathcal{A}}\pi(a|s)\sum_{s'\in\mathcal{S}}P(s,a,s')(R(s,a,s')+\gamma V_\pi(s')).$$

For each state, this is equal to the probability of eventually scoring. This probability can also be computed empirically from the data by identifying possession sequences that lead to a goal later on. Averaged over the six teams, the mean absolute error ($\pm$ 1 std) between the results from the model and the empirical results is 0.029 ($\pm$ 0.01) using the offensive grid and 0.018 ($\pm$ 0.01) using the full regular grid. This illustrates that the models provide fairly accurate estimates of derived results. The small difference can be attributed to the fact that the Markov model approach generalizes over all possible sequences to reach the goal, whereas not all sequences are ever demonstrated by the team in the data set.

6. Use cases

We illustrate the practical use of the proposed model on three in-game decision making use cases. The first two use cases focus on analyzing long-distance shooting behavior (i.e., shots from outside the penalty box). Figure 3 illustrates that the frequency of these shots has declined in the EPL in recent seasons. This is possibly driven by the insight from soccer analytics that a few close-by shots will yield more goals than many long-distance shots. Hence, it seems that players are forgoing long-shots in the hopes of generating a better shot later on. However, long-distance shots may still be beneficial as immediately shooting at least ensures the chance of scoring. Forgoing them might yield a higher quality shot later on, but there is no guarantee of this ever happening as the team might lose the ball along the way. Therefore, in the first use case, we investigate in which long-distance locations immediately shooting is preferred over moving once prior to shooting. In the second use case, we assess what the effect would be if a team modified its long-distance shooting behavior. These experiments are similar to the ones in Van Roy et al. 2021. The third use case focuses on how the model can be used to value the riskiness of each player's actions.

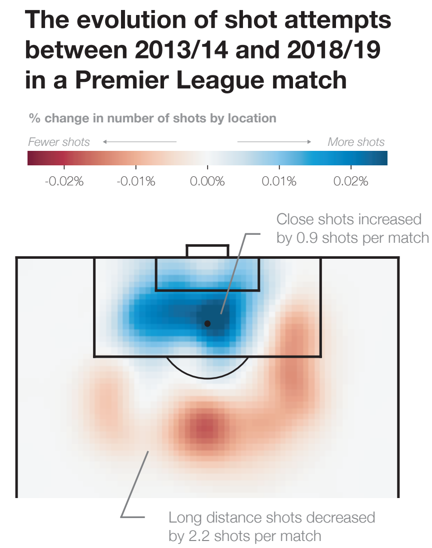


Figure 3: Illustrates the evolution of shooting in the EPL between the 13/14 and 18/19 seasons. Image from Van Roy et al. 2021.

6.1. Shoot or move

To investigate when shooting from long-distance might be preferred over moving once prior to shooting, we evaluate for each long-distance shooting location (Figure 1a) which option would result in higher chances of scoring. To compute these probabilities, we employ the approach of Van Roy et al. 2021 which uses techniques from probabilistic model checking. These techniques allow to formulate certain properties as temporal logic formulae and evaluate them against the model. Specifically, we model the probability of each choice (i.e., shoot immediately and move once before shooting) resulting in a goal as a temporal logic formula and use PRISM (Kwiatkowska et al., 2011) to evaluate them.[5]

Figure 4 shows the results for Chelsea, Huddersfield, and Manchester City. The teams have similar good shooting locations with slight differences. This is likely due to shoot-

[5]See https://www.prismmodelchecker.org/. We used PRISM v4.5.

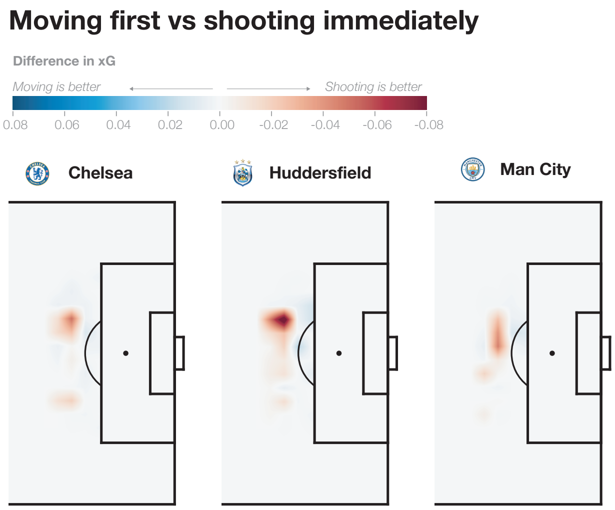


Figure 4: Long-distance locations from which shooting (red) and moving prior to shooting yields higher chances of scoring (blue).

Goal difference between the adjusted and the original policy

More long distance shots →

	+ 5%	+ 10%	+ 20%
Chelsea	0.6	1.3	2.5
Huddersfield	0.4	0.8	1.6
Man City	0.4	0.8	1.6
Man Utd	0.4	0.8	1.6
Newcastle	0.3	0.6	1.2
Burnley	0.2	0.4	0.7

Figure 5: Effect on the expected number of goals for each team over one season when the team shoots more often from the identified good shooting locations.

ing from distance being uncommon, and the hierarchical modeling approach thus shrinking the probability model of each team to the global prior. The differences do show that the good shooting locations lie both on the left and right side of the penalty arc for Chelsea, more on the left side for Huddersfield, and more towards the center in front of goal for Manchester City. Even though deciding whether to shoot or not is a split-second decision, being able to use this knowledge during training and match preparation might improve a team's performance.

6.2. Counterfactual policy analysis

To evaluate whether shooting more often from the previously identified shooting locations would be beneficial, we estimate the expected number of goals when using such a counterfactual policy. We adapt the team's original policy and increase the probability of shooting in the good shooting locations while redistributing the remaining probability mass over all other actions in those states according to their original share. Additionally, a change to the shot success probabilities in these states is also needed because adapting a team's frequency of shooting will naturally also affect the likelihood of converting those shots. When teams increase their propensity of shooting, they will most likely take more sub-optimal shots, which are of lower quality. Therefore, we will value each additional shot as follows (Van Roy et al., 2021):

$$P(s, shoot, goal) - (\mu_s - \mu_s^{low}).$$

Here, μ_s stands for the average xG (i.e., expected goals) value of all shots taken in state s and μ_s^{low} stands for the average of the xG values of all below-average shots in state s. The xG values of all shots can be found in the data set and signify the probability of converting each specific shot. Finally, estimating the expected number of goals under this policy is done by using the fundamental matrix to estimate the expected number of visits to each state $s \in \mathcal{L}$, calculating the expected number of shots in each state, and valuing each shot according to its success probability.

Figure 5 illustrates the results of adapting each of the six teams' policies. Each team would be expected to score at least one extra goal over the course of a season when shooting 20% more often from the long-distance locations identified in Section 6.1. For Chelsea, increasing their long-distance shot volume is expected to add 2.5 additional goals. Given the rarity of goals, one or two extra goals can have a large impact. Each goal scored equates to one point in the final league table (Figure 6). With teams sometimes ending within mere points from each other, shooting more often from good long-distance zones could save them from relegation or allow them to win the competition. While this only provides an estimate on the expected effect with all things being equal, it could give practitioners a better insight into which changes to a team's tactics might prove fruitful.

6.3. Rating player's riskiness

Finally, we illustrate how these models can be used to value the riskiness of a player's decision making. Valuing a player's actions is typically done based on the action's contribution to scoring (Decroos et al., 2019). While this is very useful to determine the contributions of offensive players, it does not allow for objectively assessing their decision making abilities. With the intended end location of actions in our model, this does become possible. We use the data of the 17/18 EPL season to train the models (using the regular

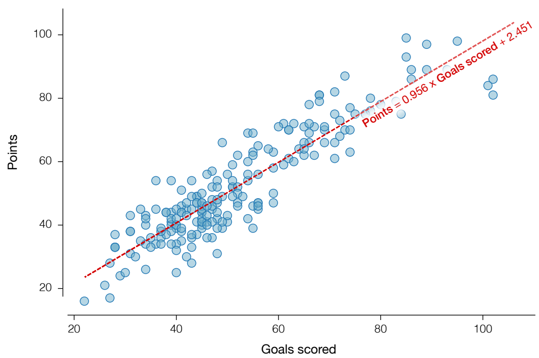


Figure 6: Obtained points in the final table as a function of the total number of goals scored using data of the 10/11 throughout the 19/20 EPL seasons. Image from Van Roy et al. 2021.

grid) and value the riskiness of the movement actions of the 18/19 EPL season.

The riskiness of a player's actions can be valued as follows using the team-specific model of the player's team. In the case of a successful action i that moves from state s to state s', the action is awarded a risk score equal to:

$$risk\_score_i = \beta_i - P(s, move\_to(s'), s')$$

where β_i denotes the average success probability of all other choices that could have been made. In the case of a failed action i, which moves from state s to a set of intended states I, the action is awarded a risk score equal to:

$$risk\_score_i = \beta_i - \sum_{s' \in I} P(s, move\_to(s'), s') * \rho_i^{s'}$$

where β_i again denotes the average success probability of all other choices and $\rho_i^{s'}$ denotes the probability of s' being the intended end location of action i. The risk scores of each player's actions are then summed and normalized per 90 minutes of game time.

Table 3 shows for the 18/19 EPL season the top-10 most risky players who play for one of the six considered teams and have at least played two full games. Table 4 shows the top-10 least risky players. A first notable result is that no player received a positive net risk rating, meaning all players perform more conservative actions than they do risky ones. This is a good sign as players who mostly play risky will not likely be picked by managers. Secondly, the top-10 least risky players features more top players than the top-10 most risky players. This could be due to the fact that these players have more experience and therefore make more optimal (less risky) choices. However, there is also a clear split visible between the different teams. This indicates that teams with a more possession-based style (e.g., Manchester City) are picked up as less risky, whereas teams that play many long balls and have a direct style of play (e.g., Burnley) can be seen as more risky. Both tables also feature a mix of attackers, defenders, and midfielders, and show a slight preference for attacking players.

Table 3: Top-10 most risky players who played at least two games in the 18/19 EPL season with their corresponding risk rating.

R_{risk}	Team	Player	Rating
1	Newcastle	Jacob Murphy	-7.697
2	Burnley	Ashley Barnes	-8.407
3	Burnley	Sam Vokes	-9.366
4	Burnley	Chris Wood	-9.371
5	Burnley	Matej Vydra	-9.625
6	Burnley	James Tarkowski	-9.687
7	Burnley	Kevin Long	-10.005
8	Newcastle	Yoshinori Muto	-10.162
9	Huddersfield	Laurent Depoitre	-10.381
10	Burnley	Aaron Lennon	-10.741

Table 4: Top-10 least risky players who played at least two games in the 18/19 EPL season with their corresponding risk rating.

R_{safe}	Team	Player	Rating
1	Man City	Oleksandr Zinchenko	-50.107
2	Man City	Fabian Delph	-45.178
3	Man City	Danilo	-44.134
4	Man City	John Stones	-43.649
5	Man City	David Silva	-43.621
6	Man City	Ilkay Gundogan	-43.217
7	Man City	Nicolas Otamendi	-42.887
8	Man City	Aymeric Laporte	-42.293
9	Chelsea	Mateo Kovacic	-41.760
10	Man City	Vincent Kompany	-40.915

7. Conclusion

In this work, we have focused on learning and analyzing a Markov Decision Process from soccer event stream data. We showed how the intended end locations of actions can be learned using a combination of predictive modeling and domain knowledge, and that a hierarchical Bayesian approach to learning the transition model can be used to mitigate sparsity issues. We provided intermediate evaluations and outlined an approach to evaluate the final model. The use cases illustrated that these models can be used both for evaluating and rating players. The insights of this can be used during training sessions and match preparation to aid in-game decision making, and for recruiting new players.

Acknowledgments

This work was supported by the Research Foundation – Flanders under EOS No. 30992574 and the KU Leuven Research Fund (C14/17/070, C14/18/062).

References

Bellman, R. Dynamic programming. *Science*, 153 (3731):34–37, 1966. ISSN 0036-8075. doi:

10.1126/science.153.3731.34. URL https://science.sciencemag.org/content/153/3731/34.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950. ISSN 0027-0644.

Cervone, D., D'Amour, A., Bornn, L., and Goldsberry, K. A multiresolution stochastic process model for predicting basketball possession outcomes. *Journal of the American Statistical Association*, 111(514):585–599, 2016. ISSN 0162-1459.

Decroos, T., Bransen, L., Van Haaren, J., and Davis, J. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1851–1861, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6201-6. doi: 10.1145/3292500.3330758.

Fernández, J., Bornn, L., and Cervone, D. A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions. *Machine learning*, 2021. ISSN 0885-6125.

Goldner, K. A markov model of football: Using stochastic processes to model a football drive. *Journal of Quantitative Analysis in Sports*, 8:1–18, 2012.

Hirotsu, N. and Wright, M. Using a markov process model of an association football match to determine the optimal timing of substitution and tactical decisions. *Journal of the Operational Research Society*, 53(1):88–96, 2002. ISSN 0160-5682.

Kwiatkowska, M., Norman, G., and Parker, D. PRISM 4.0: Verification of probabilistic real-time systems. In Gopalakrishnan, G. and Qadeer, S. (eds.), *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pp. 585–591. Springer, 2011.

Liu, G. and 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. doi: 10.24963/ijcai.2018/478. URL https://doi.org/10.24963/ijcai.2018/478.

Liu, G., Luo, Y., Schulte, O., and Kharrat, T. Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining and Knowledge Discovery*, 34, 09 2020. doi: 10.1007/s10618-020-00705-9.

Luo, Y., Schulte, O., and Poupart, P. Inverse reinforcement learning for team sports: Valuing actions and players. In Bessiere, C. (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 3356–3363. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/464. URL https://doi.org/10.24963/ijcai.2020/464. Main track.

Routley, K. and Schulte, O. A markov game model for valuing player actions in ice hockey. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pp. 782–791, 2015.

Rudd, S. A Framework for Tactical Analysis and Individual Offensive Production Assessment in Soccer Using Markov Chains. In *New England Symposium on Statistics in Sports*, 2011. URL http://nessis.org/nessis11/rudd.pdf.

Sandholtz, N. and Bornn, L. Replaying the nba. In *Proceedings of the 12th Annual MIT Sloan Sports Analytics Conference*, 2018.

Sandholtz, N. and Bornn, L. Markov decision processes with dynamic transition probabilities: An analysis of shooting strategies in basketball. *The annals of applied statistics*, 14(3):1122–1145, 2020. ISSN 1932-6157.

Schulte, O., Khademi, M., Gholami, S., Zhao, Z., Javan, M., and Desaulniers, P. A markov game model for valuing actions, locations, and team performance in ice hockey. *Data Mining and Knowledge Discovery*, 31(6): 1735–1757, 2017. ISSN 1384-5810.

Singh, K. Introducing expected threat. https://karun.in/blog/expected-threat.html, 2019.

Van Roy, M., Robberechts, P., Decroos, T., and Davis, J. Valuing on-the-ball actions in soccer: A critical comparison of xT and VAEP. In *Proceedings of the 2020 AAAI Workshop on AI in Team Sports*, 2020.

Van Roy, M., Robberechts, P., Yang, W.-C., De Raedt, L., and Davis, J. Leaving goals on the pitch: Evaluating decision making in soccer. In *Proceedings of the 15th Annual MIT Sloan Sports Analytics Conference*, 2021.

Wang, J., Fox, I., Skaza, J., Linck, N., Singh, S., and Wiens, J. The advantage of doubling: A deep reinforcement learning approach to studying the double team in the nba. In *Proceedings of the 12th Annual MIT Sloan Sports Analytics Conference*, 2018.

Yam, D. Attacking contributions: Markov models for football. https://statsbomb.com/2019/02/attacking-contributions-markov-models-for-football/, 2019.

A. EXPERIMENTAL SETUP

We performed all experiments in this paper in Python. We provide the details of our setup below:

Gradient boosted trees ensembles To optimize the hyperparameters of the XGBoost models, for each action type (i.e., pass, cross) we performed 5-fold cross-validation randomized search on the training set (70% of the data set of the specific action type) with 50 parameter settings sampled from the following distributions:

```
'max_depth': random integer [2, 10],
'min_child_weight': random integer [1,11],
'gamma': uniform [0,1],
'reg_alpha': uniform [0,1],
'reg_lambda': uniform [0,10],
'base_score': uniform [0.1,1],
'subsample': uniform [0.5,1],
'colsample_bytree': uniform [0.5,1],
'colsample_bylevel': uniform [0.5,1]
```

The other hyperparameters were set to their default values.

The models were trained and evaluated on a computing server running Ubuntu 18.04 with 16GB of RAM and an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz. Training the ensemble on passes (crosses) for one team using the offensive grid takes approximately 5.7 (3.6) minutes, and 2 (1.3) minutes when using the full regular grid.

Bayesian models Apart from the parameter settings mentioned in the main text, all other parameters were set to their default values.

The models were trained and evaluated on a computing server running Ubuntu 18.04 with 128GB of RAM and an Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz. Training the policy (transition) model for one team using the offensive grid takes approximately 15 (1.5) hours, and 4 hours (32 minutes) when using the full regular grid.