Exploring the application of soccer mathematical models to game generation on a simulated environment

Juan Carlos Núñez, Bruno Dagnino**

Metrica-Sports
www.metrica-sports.com

November, 2020

Abstract

In recent years there have been various models that use historical data and the Markovian characteristics of the game of soccer to model the game. Some of the most referred to models are pitch control, expected possession value and expected goals. However, while such a significant progress has been made on modeling the game, very little has been done to date in terms of simulating it. In this paper we take advantage of the currently ongoing Kaggle competition promoted by Google Research and Manchester City, to explore for the first time how these models could help generate simulated game actions at a competitive level. We approach the problem by proposing a weighted combination of these models and by characterizing the xG of the game engine. Our agent proved to be able to generate realistic game plays and to compete at a very high level, achieving at the moment of the writing of this paper the top 1.5% of competitors (position #13 out of 813)

Keywords: automatic, agent, game play, pitch control, expected possession value, expected goals,

1. Introducction

In the past years there has been a growing interest in the field of sports analytics to try to model and understand the game of soccer from an analytical perspective. An approach to the subject that has gained traction in recent years is to model the game of soccer as a Markov process [1][2], in which what's gonna happen on time $t+1$ only depends on the state of the system at time t. Some of the most successful and well-known models based on this

*Code can be found here: https://www.kaggle.com/jcnunezmoreno/epv-pc-xg-metrica-release

**Corresponding author

Email address: bruno@metrica-sports.com (Bruno Dagnino)

approach are pitch control (PC) [3], expected possession value (EPV) [4] and expected goals (xG) [5]. PC models estimate the probability that either team has of achieving possession of the ball should it be played at a certain location on the field. EPV in its simplest approach estimates what is the likelihood of a team scoring a goal given it's in possession of the ball at a certain coordinate (x, y). xG models how likely a player is to score a goal if he shoots from a location (x, y). There are other ways to model concepts related to the three described above (like pass probability map, xT, VAEP, g+) [6], but for the purposes of this paper we are going to focus on these three.

The progress so far in this research area has been done mostly by computing models based on historical data. There has been very little progress or research done on simulating the game of soccer. However that is quickly changing. On July $2019th$ Google Research released the Google Research Football Environment (GRFE) [7], a novel RL environment where agents aim to master the world's most popular sport—football. Moreover Manchester City F.C and Google Research have recently announced a Kaggle competition [8] using the GRFE. The researchers want to explore the ability of artificial intelligent (AI) agents to play in complex soccer environments. The goal is to open up an environment for the community to foster progress in the generation and understanding of the game of soccer.

In GRFE, each moment (or time step t) the agent gets an observation from the system status which has the information about what the ball and players are up to. The agent has to decide which of the 19 actions should it take: go this way or that way, start sprinting, pass short or long, or try a shot. The agent can be implemented by a basic rule-based algorithm, where conditions like *"shoot if you are close to the goal"* or *"go to the ball if you don't have it"* can be developed. But writing out tactical instructions based on rules (like if-then statements) is extremely tedious and limited.

However, the fact that the game of soccer can be considered a Markovian system makes it a great problem to apply reinforcement learning (RL)[9] methods on. While this approach has a lot of potential, there are different drawbacks that can make it complicated to train an RL system in an environment as complex as the one that takes place on a soccer game. On recent successful RL projects, even training agents to play simple video games require substantial computational resources that may not be available to a large fraction of researchers due to combining hard games, long episodes, and high-dimensional inputs [7]. In addition, an optimized policy through the goal reward in the training phase can cause it to take paths that make no sense from the point of view of soccer analytics understanding. That is, the system could find an optimal way just by running to the goal, and shooting from outside the area. This could make AI always win games, but from a tactical point of view of the game, it would not add much to the the understanding of the game. As with other famous examples in the history of RL [10][11], the trained agent could exploit the peculiarities of the environment, rather than the dynamics of the game itself.

This aspect of the problem is what leads us to the line of inquiring we are gonna address in this paper: *How realistic is this environment? Would the concepts that were developed based on real world data (PC, EPV, xG) be effective in the GRFE environment as well? Could we use these models to make decisions on what an agent should do? How well can we perform on the competition versus other users taking an RL approach?*

In this work we are going to explore the effectiveness of the mathematical models already proposed in the literature, as PC, EPV or xG, for the automatic generation of the game. We will study whether it's possible to develop an agent based on these concepts without RL methods and avoiding as far as possible rule-based algorithms.

The main contributions of this work are then summarized as follows:

- We implemented an agent that generates automatic game play based on a weighted combination of PC, EPV and xG.
- We have built an xG map for the GRFE.
- We demonstrated competitive results in the Kaggle competition.

Our work is not a complete exploration of the best way to combine these models to compete on the GRFE, but rather a first exploratory approach to the topic. Besides exploring a new avenue of research, this approach could be used for example as a seed in the RL systems, providing a way to avoid having to train from scratch (computationally expensive) while at the same time creating styles of play that capture the essence of the game

2. Our proposal

We approach the problem as a discrete optimization one, in which the goal is to have a player with the ball in the best possible position to score a goal. For this purpose, the agent will be selecting the action that offers the greatest benefit in terms of player possession and location. The agent takes an observation at the current environment $state(t)$ and outputs an action from the available set in this time step (t). Figure 1 shows an overview of the system, which is divided in two main blocks, the environment manager, which is responsible for preparing and preprocessing the observation, and the decision module, which selects an action based on the current observation and the PC, EPV and xG models.

2.1. Environment manager

This module processes the state observation in order to provide the decision module data it can understand. First, the positions of the players and the ball are projected in a future time step $(t + n)$ using orientation vectors and current positions (the reason for

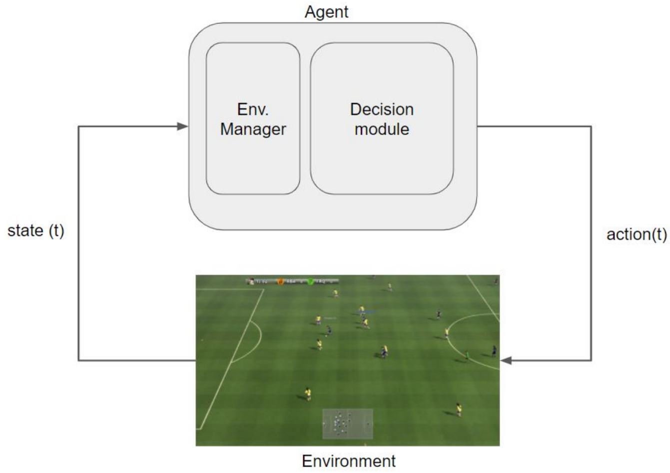


Figure 1: System overview

this projection is that the environment takes a few time steps to execute the action sent). Then, players and ball coordinates are normalized (-0.5, 0.5). Finally, the attacking or the defending phase is selected based on the team possession.

2.2. Decision module

The version of the GRFE on which the Kaggle competition is based allows you to control one player per team at the time, both during attacking (attacking mode) and defensive (defending mode) phases of the game. When attacking you can control the player in possession of the ball, whilst while defending you can control the player closest to the ball. Our approach on this paper only affects the actions taken during the attacking phase. While defending, the closest player to the ball just runs towards it to try to recover it.

The aim of the attacking phase is to score a goal. To do that, the attacking team should have a player in possession of the ball in an optimal position to shoot and score. This position can be reached through passing to other players or running with the ball. When the player reaches the location, the shot can be executed. The suitability for the shot can be related to several factors such as the distance/angle to the goal, the distance/angle to the goalkeeper, the defending team situations, among others. Whether the right decision is to make a pass or run in a certain direction, can be related to the positions of the players of its own team and the opponent, or the value of the space they occupy. These variables have been modelled by PC, EPV and xG models.

The proposed agent will optimize the action selection through the cost maximization of a weight function which combines PC, EPV and xG models, in order to bring a player with the ball to an optimal position. In Figure 2 the agent overview for the attacking mode is shown. At each time step (t) of the system loop, the agent evaluates all available options

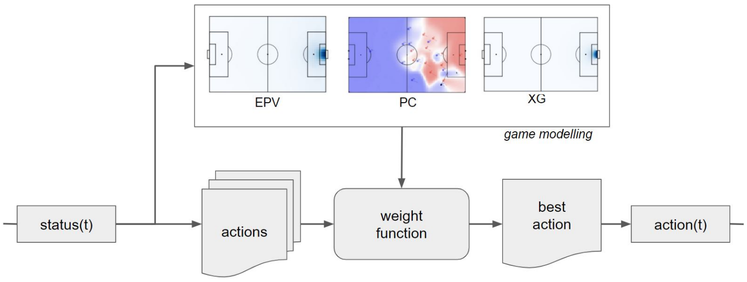


Figure 2: Agent overview for the attacking mode

that the player has (actions), then, the weight function scores them and executes the best one.

Actions set generation

At each time step (t) our agent will generate twelve action candidates: (1) *move right*, (2) *move top right*, (3) *move top*, (4) *move top left*, (5) *move left*, (6) *move down left*, (7) *move down*, (8) *move down right*, (9) *long pass*, (10) *short pass*, (11) *long high pass* and (12) *shot*. It's important to notice that within the action set of the environment you can't choose which player to pass, or to which location in the field to pass. The only pass related actions you can choose from are 9, 10 and 11 and the system will make a pass to the player that is at the closest angle with respect to the orientation of the body of the player in possession. For this reason, only that player is taken into account at each time step to decide whether to carry, pass or shoot.

Weight function

To evaluate all possible actions the PC, EPV and xG need to be computed at the positions where the action will place the player with the ball (either by moving to, or by receiving the ball in that position). Let's assume the state of the Figure 3. Controlled player (red dot) is oriented towards the receiver player (green). The agent can move the player with the ball to each of the eight directions (orange dot), and it could also make a pass to the receiver player (who will be the controlled player after receiving the pass).

To compute the best action to take, we proposed a weight function that gives each one of the 9 options a score, so the agent will weigh each position where the controlled player will be (orange and green dots) following the Equation 1. Where $w(t, A_i)$ is the weight of action A_i at time t. $PC(t, x_i, y_i)$ is the pitch control value at the position x_i, y_i where the action will move the player (or receiver player in the case of a pass) at time t. $EPV(t, x_i, y_i)$ and $xG(t, x_i, y_i)$ are the expected possession value and expected goal respectively at time step t in the same x_i, y_i position. Let's be α, β, γ factors which weigh the relevance of each model.

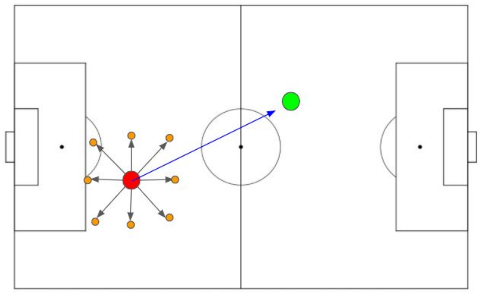


Figure 3: State example for action generation

$$w(t, A_i) = \alpha * PC(t, x_i, y_i) * EPV(t, x_i, y_i) + \beta * EPV(t, x_i, y_i) + \gamma * xG(t, x_i, y_i) \qquad (1)$$

The proposed weight function is composed of a weighted sum of the model value at a given location. PC value is weighted by EPV in order to put both models in the same order of magnitude for small values of the EPV. The values of α, β, γ factors have been estimated empirically. For PC and EPV models we have used the ones available at Lauries Shaw's repository [12] as part of the Friends of Tracking initiative.

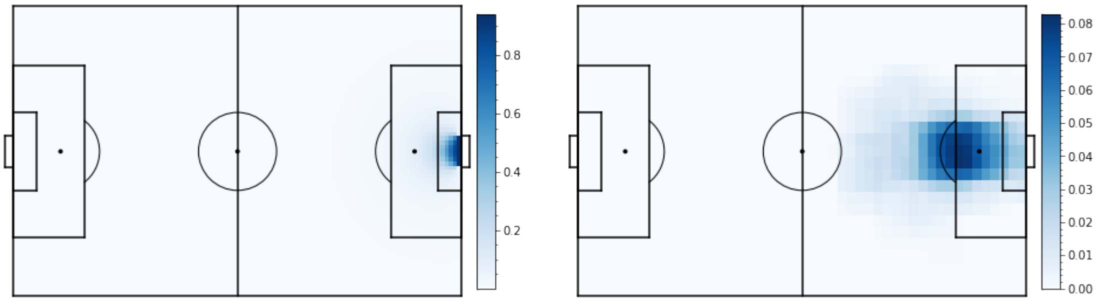
If a pass action is selected as the best action to take, the pass type will be selected based on the distance to the receiving player and the presence or not of an opponent in the line of the pass. A shot action will be performed if the controlled player has a $xG(t, x_i, y_i)$ value higher than a threshold or alternative if the goalkeeper is close to the controlled player. Pass actions will be avoided if there is not a best oriented player from its own team.

2.3. Environment expected goal

For the initial development of the agent we have employed the basic expected goal model [13] (Figure 4a). Visual exploration of the first game suggested that the agent wasn't making the best decision regarding when to shoot. Based on this observation, we computed an approximation of the xG model of the GRFE (Figure 4b).

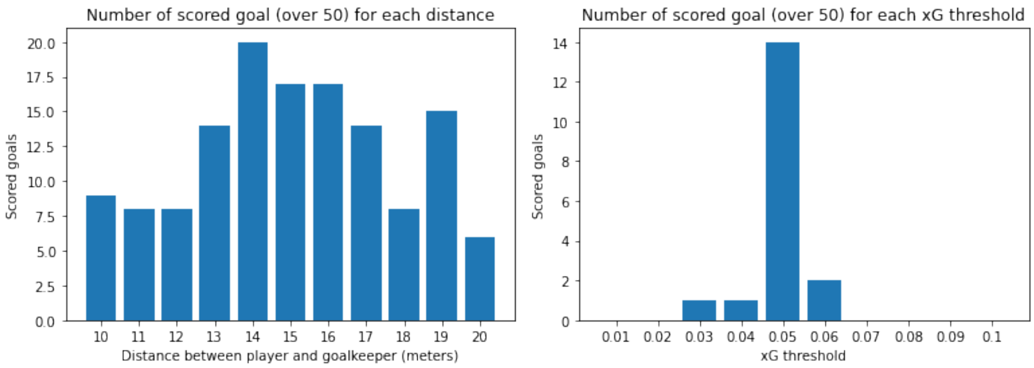
To do that we have simulated a number of shots (100) of the player from different positions on the field. The positions have been extracted from a grid of 100x50 divisions, so the resolution is approximately one square meter, taking into account the measurements of the field are 106x68m. The goalkeeper is always on the goal and controlled by the environment, so this model also encoded its behaviour.

Besides the xG of a particular location, we observed that the distance to the goalkeeper at the moment of the shot could have a great impact as well. To study that, we have launched a number of runs with a grid of parameters calculating the total reward for each



(a) (b)

Figure 4: Basic xG model (a), and the computed approximation of the xG model for GRFE



(a) (b)

Figure 5: Optimal shooting distance (a) and xG threshold

one. As can be seen in the Figure 5a, the total number of scored goals over 50 shoot for each distance (between player and goalkeeper) is shown. Where it can be determined that the optimal shooting distance is 14 meters. We did the same for the xG threshold finding an optimal solution at 0.05 (Figure 5b).

3. Experimental Results

The Kaggle IA soccer competition allows us to submit our agent and compare it with other agents in an objective way. Each submission will play episodes (games) against other agents on the ladder that have similar skill ratings. Over time, skill rating will go up with wins or down with losses. Every agent submitted will continue to play games until the end of the competition. More details about the evaluation can be found in the challenge evaluation section [14]. The current score of our proposal can be found in the leaderboard [15]. At the

Figure 6: Example of a piece of game of the proposed agent

moment of the writing of this article (with 1 months still to go till the end) we are on the top 1.5% of competitors (position #13 out of 813). It should be noted that the main objective of this work is not to achieve first place, but to generate interesting game style based on tactical concepts. Nonetheless the proposed system obtains very competitive results.

Beyond the results, it is interesting to make a qualitative evaluation of the proposed method in order to check if the agent works as expected. In the Figure 6, a play from the proposed agent is shown. As it can be seen, the controlled player moves to a position where there is a greater chance of a goal, passing to another player when it is in a better position. In this video [1], it can be seen that our team (left team) has learned to score a goal through passing, while most agents still dribble straight for goal. In this other video [2] we present some game examples from our agent.

4. Conclusions

In this paper we presented an algorithm for the automatic soccer game generation. Our proposal is based on a discrete optimization of the actions, weighted by a combination of PC, EPV and xG models. We demonstrate the effectiveness of these mathematical models, well-known in sports analytics, to build an agent obtaining competitive results in the Kaggle challenge. So far, we are in the top 1.5% of competitors (position #13 out of 813) teams. In addition, we have built the xG model of the environment, which has allowed us to significantly improve the scoring percentage of our agent.

To the extent of our knowledge, this is the first work to explore the application of existing models on the field of soccer analytics to GRFE, as well as to the simulation of soccer games in general. While innovative, since it was based on a recently announced Kaggle competition this work is far from being a complete exploration of the best ways to combine these types of models. Rather, it should be considered a starting point for research in the game generation through soccer analytics concepts. There are many lines of future research that could expand

[1] https://vimeo.com/472237573

[2] https://vimeo.com/472237690

on the initial findings and approach presented here. Different weights functions could be explored. Weights α, β, γ could be programmatically explored. EPV, PC and improved xG specific to the GRFE could be computed. Different EPV models and weight functions could be explored to generate different styles of play. While this approach is not RL based, the game actions generated by this approach could be used as seed to train it, to help avoid the limitations of the environment and help generate tactile sounding playing styles.

As a final thought, while limited and far from hyper realistic, the existence of environments like the GRFE and of competitions like the ones currently on Kaggle, are a great experimentation ground for the field of soccer analytics. In the real world, we are constrained by the actions players took, and by the games teams played. Whereas in an environment like the GRFE you can simulate thousands of games on which players take the actions you specify. This allows for infinite experimentation, like for example, combining different models like PC, EPV and xG to create game actions form them, rather than just observing the results that took place on a real game.

It's important to notice that the applications of this line of research could have a great impact in real world soccer as well. For example, similar to how chess engines are used nowadays, more advanced simulations could be used to simulate games between different teams with different tactics. Or study the influence in results of starting one player vs another, or playing one system vs another one. It could also serve as a way to cross examine models. You could derive models from RL approaches to the virtual environment and apply them to the real world, and vice versa, with more realistic engines you can validate real world models in a virtual environment.

While still mostly unexplored, we believe experimentations in the line of the approach proposed on this paper will have a great impact on the field of soccer analytics.

References

[1] P. A. Gagniuc, Markov Chains: From Theory to Implementation and Experimentation, John Wiley Sons, 2017. doi:10.1002/9781119387596.
URL https://doi.org/10.1002/9781119387596

[2] Markov chain: Definition of Markov chain in US English by Oxford Dictionaries, Oxford University Press, 2017, english.

[3] W. Spearman, A. Basye, G. Dick, R. Hotovy, P. Pop, Physics-based modeling of pass probabilities in soccer, MIT Sloan, Sports Analytics Conference (2017).

[4] J. Fernández, L. Bornn, D. Cervone, Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer, MIT Sloan, Sports Analytics Conference (2019).

[5] W. Spearman, Beyond expected goals, MIT Sloan, Sports Analytics Conference (2018).

[6] D. Sumpter, Mathematical modelling of football, https://www.youtube.com/watch?v=$_{UQ}5jpeGMo$.

[7] K. Kurach, A. Raichuk, P. Stanczyk, M. Zajac, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, S. Gelly, Google research football: A novel reinforcement learning environment, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Sym-

posium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 4501–4510.
URL https://aaai.org/ojs/index.php/AAAI/article/view/5878

[8] Kaggle, Google research football with manchester city f.c., https://www.kaggle.com/c/google-football/overview.

[9] R. S. Sutton, A. G. Barto, Reinforcement learning - an introduction, Adaptive computation and machine learning, MIT Press, 1998.
URL https://www.worldcat.org/oclc/37293240

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nat. 529 (7587) (2016) 484–489. doi:10.1038/nature16961.
URL https://doi.org/10.1038/nature16961

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, D. Hassabis, Mastering chess and shogi by self-play with a general reinforcement learning algorithm, CoRR abs/1712.01815. arXiv:1712.01815.
URL http://arxiv.org/abs/1712.01815

[12] LaurieOnTracking, Pitch control and expected possession models, https://github.com/Friends-of-Tracking-Data-FoTD/LaurieOnTracking (2020).

[13] D. K. Leuven, Soccer xg, https://github.com/ML-KULeuven/soccer_xg (2020).

[14] Kaggle, Google research football with manchester city f.c. evaluation process, https://www.kaggle.com/c/google-football/overview/evaluation.

[15] Kaggle, Google research football with manchester city f.c. leaderboard, https://www.kaggle.com/c/google-football/leaderboard.